1982
COMPLETED
152

Y 3. N88:25/1233/v.4

NUREG/CR-1233, Vol. IV
UCID-18146

The Structured Assessment Approach Version 1 Computational Analysis Package

Volume IV

A. A. Parziale, C. J. Patenaude, P. A. Renard,
D. W. Freeman, D. J. Ross
I. J. Sacks, Analytic Information Processing

Prepared for
U.S. Nuclear Regulatory Commission

Lawrence
Livermore
National
Laboratory

I

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

Available from

GPO Sales Program
Division of Technical Information and Document Control
U. S. Nuclear Regulatory Commission
Washington, D. C. 20555

Printed copy price: $7.50

and

National Technical Information Service
Springfield, Virginia 22161

NUREG/CR-1233, Vol. IV
UCID-18146
RS

The Structured Assessment Approach Version 1 Computational Analysis Package

Volume IV

Manuscript Completed: October 1979
Date Published: April 1982

Prepared by
A. A. Parziale, C. J. Patenaude, P. A. Renard,
D. W. Freeman, D. J. Ross
I. J. Sacks, Analytic Information Processing

Lawrence Livermore National Laboratory
7000 East Avenue
Livermore, CA 94550

Prepared for
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
NRC FIN No. A0115

NOTICE

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 1717 H Street, N.W.
   Washington, DC 20555
2. The NRC/GPO Sales Program, U.S. Nuclear Regulatory Commission,
   Washington, DC 20555
3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC Office of Inspection and Enforcement bulletins, circulars, information notices, inspection and investigation notices; Licensee Event Reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the NRC/GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, and NRC booklets and brochures. Also available are Regulatory Guides, NRC regulations in the *Code of Federal Regulations,* and *Nuclear Regulatory Commission Issuances.*

Documents available from the National Technical Information Service include NUREG series reports and technical reports prepared by other federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal and periodical articles, and transactions. *Federal Register* notices, federal and state legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free upon written request to the Division of Technical Information and Document Control, U.S. Nuclear Regulatory Commission, Washington, DC 20555.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, 7920 Norfolk Avenue, Bethesda, Maryland, and are available there for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

FOREWORD

This report is Volume 4 of a four—volume document, and presents the computational analysis package which performs the Structured Assessment Approach, SSA. This volume contains both technical descriptions of each portion of the SAA and the corresponding computer program listings. This code is presently implemented on the Lawrence Livermore National Laboratory OCTOPUS computer system and uses a CDC 7600 for data processing.

The four—volume document has the following organization:

Volume I:	The Structured Assessment Approach Version 1 Volume I: Analysis Package, (Executive Summary).
Volume II:	The Structured Assessment Approach Version 1, Volume II: The License Submittal Document Content and Format for Material Control and Accounting Assessment, (Input data description required to perform safeguards assessment).
Volume III:	The Structured Assessment Approach Version 1, Volume III: Applied Demonstration of Output Results, (A description of typical assessment output results for a hypothetical fuel cycle facility described in Volume II).
UCID 18146:	Computational Analysis Package: The Structured Assessment Approach Version 1, (A description of the Version 1 computational computer programs including program listings, which were used to generate analysis output results described in Volume III from hypothetical input facility information described in Volume II).

TABLE OF CONTENTS

FOREWORD iii

ABSTRACT xiii

EXECUTIVE SUMMARY xv

1.0 INTRODUCTION 1

  1.1 Technical Overview of the Structured Assessment Approach . . 1

    1.1.1 Level 1 - Coverage 2
    1.1.2 Level 1.5 - Weak Collusion 3
    1.1.3 Level 2 - Adequacy 3
    1.1.4 Level 3 - Sensitivity 3
    1.1.5 Level 3.5 - Strong Collusion 4
    1.1.6 Level 4 - Tampering Vulnerability 4

  1.2 Overview of the Structured Assessment Approach Codes 4

2.0 STRUCTURED ASSESSMENT APPROACH TECHNICAL DESCRIPTION AND CODES . . 7

  2.1 License Submittal Document 7

    2.1.1 NRC Control 8
    2.1.2 Code Listing 8

  2.2 License Submittal Document Preprocessor 9

    2.2.1 Structure of the LSD Preprocessor 11
    2.2.2 Code Listing 13

  2.3 Core Preprocessor 13

    2.3.1 Technical Description 14
    2.3.2 Condensation 17
    2.3.3 Expansion of Access and Control Lists for Facility Personnel 28
    2.3.4 Code Listing 37

  2.4 Level 1 - Monitor Target Set Coverage 41

    2.4.1 Technical Description 41

      2.4.1.1 Area Reachability 43
      2.4.1.2 Target Selection 47
      2.4.1.3 Acquisition Target Set Generator 48
      2.4.1.4 Output Summarizer 48

    2.4.2 Sample Output 50
    2.4.3 Code Listing 50

TABLE OF CONTENTS (continued)

2.5 Level 1.5 - Weak Collusion Test 50

2.5.1 Technical Description of Level 1.5 50
2.5.2 Sample Output 55
2.5.3 Code Listing 55

2.6 Level 2/3 - Adequacy & Sensitivity Determination 56

2.6.1 Technical Description of Level 2/3 56

2.6.1.1 General Flow of Levels 2 and 3 56
2.6.1.2 Probabilty Calculation 64

2.6.1.2.1 Probability Code 64
2.6.1.2.2 Bounding Tests 68
2.6.1.2.3 Safeguard System False Alarm rate 73

2.6.2 Sample Output 75
2.6.3 Code Listing 75

2.7 Level 3.5 - Second Phase Collusion Test 76

2.7.1 Technical Description of Level 3.5 76
2.7.2 Sample Output 79
2.7.3 Code Listing 79

2.8 Level 4 - Full Tampering Vulnerability 79

2.8.1 Technical Description 80

2.8.1.1 General Technique Description 80
2.8.1.2 Material Control System Analysis 81
2.8.1.3 Material Accounting System Analysis . . . 91
2.8.1.4 Boolean Analysis Code Description - CLAMOR . 95

2.8.1.4.1 Controller Options 98
2.8.1.4.2 Data Management 98

TABLE OF CONTENTS (continued)

2.8.1.4.3 Data Structure 98
2.8.1.4.4 Getting Help 99
2.8.1.4.5 CLAMOR Initialization 99
2.8.1.4.6 Output Information 100
2.8.1.4.7 Enlarging the Work Space . . . 101
2.8.1.4.8 Adjacency Specification . . . 102
2.8.1.4.9 Path Generation 104
2.8.1.4.10 Elimination Options 105
2.8.1.4.11 Saving the Reachability Calculation 107
2.8.1.4.12 Generating Output 108
2.8.1.4.13 Solving The System 109
2.8.1.4.14 Reducing the Adjacency Information 111
2.8.1.4.15 Miscellaneous Z-Commands . . . 112
2.8.1.4.16 An Example 112

2.8.2 Sample Input 116

ACKNOWLEDGMENTS 117

REFERENCES 119

APPENDIX A: GUIDANCE TO DOCUMENTATION 121

ANNOTATED BIBLIOGRAPHY 123

LIST OF FIGURES

Figure 1: Organization of Document xvii

Figure 1.1-1: Structured Assessment Approach 1

Figure 1.2-1: Structured Assessment Approach Code Organization . . . 5

Figure 2.2-1: SAA Information Flow 10

Figure 2.3-1: Data Flow: LSDB to Level 1-3.5 15

Figure 2.3-2: Data Flow Through Core Preprocessor 16

Figure 2.3-3: Location Graph Which Requires Excessive Computation for Reachability Calculation 18

Figure 2.3-4: Adjacency Matrix for Location Graph of Figure 2.3-4 . . 19

Figure 2.3-5: Example of Unreachable Areas 21

Figure 2.3-6: Reachability Matrix for Example of Figure 2.3-5 . . . 22

Figure 2.3-7: Unmonitored Hallway Example 23

Figure 2.3-8: Adjacency Information for Example of Figure 2.3-7 . . . 23

Figure 2.3-9: Graph With Monitored Nodes Marked 24

Figure 2.3-10: Condensed Graph 24

Figure 2.3-11: Monitored Graph With Fingers 25

Figure 2.3-12: Location Traversal Graph with Monitored Nodes Marked . . 27

Figure 2.3-13: Condensed Adjacency Graph 29

Figure 2.3-14: Typical Control Bit Vectors for Levels 1.5 and 3.5 . . 28

Figure 2.3-15: Location Network Condensation 31

Figure 2.3-16: First Expansion of Access List 33

Figure 2.3-17: Recondensation for Specific Mode and Individual . . . 34

Figure 2.3-18: Expansion of Access Locations 36

Figure 2.3-19: Expansion of Individual Control List 38

Figure 2.3-20: Indirect Tamper Protection 39

Figure 2.3-21: Determination of Tamper Protection List 40

LIST OF FIGURES (Continued)

Figure 2.4-1: Monitor Target Sets 42
Figure 2.4-2: Conceptual Level 1 Flow 44
Figure 2.4-3: Typical Monitor Target Sets 45
Figure 2.4-4: Adjacency Matrix 46
Figure 2.4-5: Dominance Testing 47
Figure 2.4-6: Acquisition Target Set Generation 49
Figure 2.5-1: Example of Monitor Target Sets 51
Figure 2.5-2: Access and Control Lists 51
Figure 2.5-3: Determination of Individuals Who Can Compromise MTS . . 52
Figure 2.6-1: Example of Target Set Coverage Network 56
Figure 2.6-2: Flowchart for Level 2 and Level 3 59
Figure 2.6-3: Example Network 64
Figure 2.6-4: Conceptual Structure of Probability Algorithm 66
Figure 2.6-5: Bit Vector Representation of MTS 67
Figure 2.6-6: Probability Calculations for PROB 69
Figure 2.6-7: Detail B for PROB 70
Figure 2.6-8: Detail A for PROB 71
Figure 2.7-1: Monitor Target Set With Support Systems 77
Figure 2.8-1: General System Element 80
Figure 2.8-2: General Functional Network 81
Figure 2.8-3: Simplicity Uranium Corporation 82

LIST OF FIGURES (Continued)

Figure 2.8-4: Area Adjacency Matrix for Example 83

Figure 2.8-5: Area Graph for Example 83

Figure 2.8-6: Field-of-View Matrix 84

Figure 2.8-7: "Petri Net" for Simplicity Uranium Corporation 85

Figure 2.8-8: Area Monitoring 86

Figure 2.8-9: "Petri Net" for Simplicity Uranium Corporation with MC&A Information Flow Added. 88

Figure 2.8-10: MC&A Information Flow Adjacency Matrix 88

Figure 2.8-11: Adjacency Matrix for "Petri Net" of Figure 8 89

Figure 2.8-12: "Petri Net" for Simplicity Uranium Corporation 91

Figure 2.8-13: Higher Resolution Net for Example 93

Figure 2.8-14: Identify Late Detection Systems 91

Figure 2.8-15: Identify Components of Detection System 94

Figure 2.8-16: Identify Sources of Datas For All Components 94

Figure 2.8-17: Identify Protection for Each Data Flow 94

Figure 2.8-18: Identify Unit Models, Locations, Personnel 95

Figure 2.8-19: A Sample Network 113

LIST OF TABLES

Table 2.3-1: Super Nodes for Figure 2.3-3 26
Table 2.3-2: First Expansion of Access List 32
Table 2.3-3: Second Expanded Access List 34
Table 2.7.1: Authorized Access List 77

CONTENTS OF MICROFICHE

SAA Levels 1-3

2.1.1 NRC Control Program Code Listing 1-57

2.2.2 License Submittal Document Preprocessor Code Listing 58-66

2.3.4 Core Preprocessor Code Listing 67-98

2.4.2 Example of Summary Output for Level 1 99-101

Example of Complete Output for Level 1 102-106

2.4.3 Code Listing for Level 1 107-236

2.5.2 Example of Summary Output for Level 1.5 237-245

Example of Complete Output for Level 1.5 246-251

2.5.3 Code Listing for Level 1.5 252-298

2.6.2 Example of Level 3 Summary 299-307

Example of Level 2/3 Output Results for Adequately Covered Diversion Paths 308-312

Example of Level 2/3 Output Results for Inadequately Covered Diversion Paths 313-317

2.6.3 Code Listing for Level 2/3 318-397

SAA Level 4

2.8.2 Sample Input for Level 4 1

2.8.3 Code Listing of Level 4 2-218

ABSTRACT

A methodology, called the Structured Assessment Approach (SAA), has been developed to assess the effectiveness of material control and accounting (MC&A) safeguards systems at nuclear fuel cycle facilities. The methodology has been refined into a computational tool, the Version 1 analysis package, that has been used first to assess a hypothetical nuclear fuel cycle facility and more recently to assess operational nuclear plants.

The Version 1 analysis package is designed to analyze safeguards systems that prevent the diversion of Special Nuclear Material from nuclear fuel cycle facilities and to provide assurance that diversion has not occurred.

EXECUTIVE SUMMARY

Background

The Lawrence Livermore National Laboratory (LLNL) was tasked by the U. S. Nuclear Regulatory Commission (NRC) Office of Regulatory Research to develop a methodology for the assessment of the effectiveness of Material Control and Accounting (MC&A) safeguards systems at nuclear fuel cycle facilities. The methodology that has been developed, called the Structured Assessment Approach (SAA), was first applied in the assessment of a nuclear facility in late 1978 using "prototype" computer programs.

The development of the SAA was begun in late 1977 as one of two assessment methodologies studied. The demonstration of the results of both methodologies led the NRC to select the SAA as the technique to be developed into an applications-oriented package. The development of this integrated computational package was accelerated in May 1979, and in October 1979 the package was delivered.

Overview of SAA Version 1 Package and Report

This report presents a summary of the SAA Version 1 computational package, which is designed to analyze safeguards systems that prevent the diversion of Special Nuclear Material (SNM) from nuclear fuel cycle facilities and provide assurance that diversion has not occurred. Two distinct types of monitoring or detection systems are analyzed: fast-response-time systems designed to prevent an adversary from diverting SNM, and long-response-time systems designed to detect theft through material imbalances. The first of these systems is referred to as Material Containment Monitoring or Material Control, and the second as Material Accounting.

The analysis methodology is staged; it subjects the safeguards system at the facility to a series of increasingly stringent performance tests, ranging from a determination of whether a nontampering adversary can defeat the facility with no risk at all, to subtle questions dealing with the reliability of the detection system and the dynamics of specific diversion sequences. The advantage of the staged approach is that it allows a great deal of analysis to be done with a minimum of judgmental input from the analyst. Because each stage subjects the facility to more exacting criteria, passing a given stage does not mean that the facility safeguards system is acceptable, but failing at any point means that the safeguards system should be rejected. One of the main advantages of a staged approach is that sensitivity analyses can be performed at each stage to identify the weakest points in the system and allow safeguards system redesign. The insights gained at each stage allow the analyst to focus the detail in the next stage of the analysis on those areas where it is more likely that system problems will occur.

The SAA allows an objective and consistent analysis to be made of safeguards systems with subjective, experiential inputs made only at identifiable steps during the analysis. Analysis results from the SAA are always traceable back to licensee-submitted input data.

This Executive Summary provides a brief introduction to the SAA, its basic design philosophy, its data input requirements, and the types of outputs that result from an SAA analysis of an MC&A system.

The basic organization of this document is shown in Figure 1.

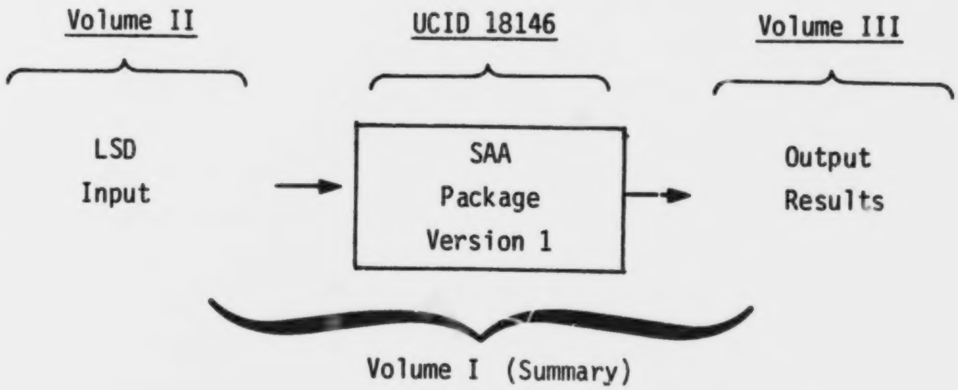


FIG. 1. Organization of documentation

Volume II presents the requirements for the data that a license applicant must submit to the NRC for analysis, along with data format requirements. The data are requested in the License Submittal Document (LSD). Volume II presents an example of the data construction process applied to a hypothetical full-scale nuclear facility. Volume II is intended to serve as an NRC user's manual to aid in the construction of the LSD for assessments in line with future upgrade rules.

A description of the assessment package output for each stage of the SAA analysis is presented in Volume III. The SAA package was applied to the hypothetical nuclear fuel cycle facility (HFCF) used as an example in Volume II. The results are discussed and interpreted in Volume III. Volume III is intended as a guide to assist the NRC license reviewer in the interpretation of SAA analysis results.

1.0 INTRODUCTION

This volume contains technical descriptions and computer code listings for all presently operational Structured Assessment Approach (SAA) Levels. The codes have been designed to run on the Lawrence Livermore National Laboratory Octopus System using a CDC 7600. Within this document the reader will find

1. A general technical overview of the SAA
2. Technical descriptions of each distinct level of the SAA
3. Code listings.

1.1 TECHNICAL OVERVIEW OF THE STRUCTURED ASSESSMENT APPROACH

Both the methodology and the conclusions from the Structured Assessment Approach are subdivided into four main levels. Two of these main levels are further divided into intermediate sublevels. The main levels are characterized by four basic models of adversary infiltration. These four main levels described below are shown schematically in Fig. 1.1-1.

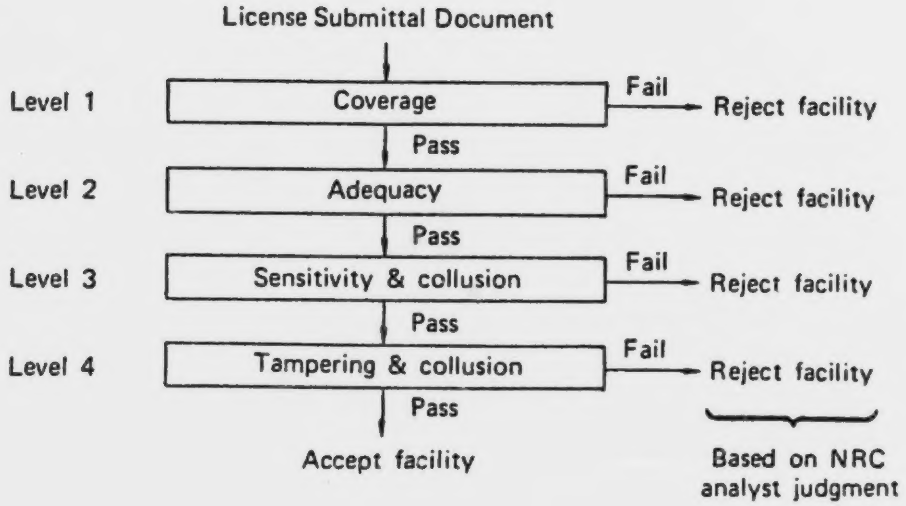

License Submittal Document
Level 1
Coverage
Fail
Reject facility
Pass
Level 2
Adequacy
Fail
Reject facility
Pass
Level 3
Sensitivity & collusion
Fail
Reject facility
Pass
Level 4
Tampering & collusion
Fail
Reject facility
Pass
Accept facility
Based on NRC analyst judgment

FIGURE 1.1-1: Structured Assessment Approach

- Level 1 -- Coverage. Can a nontampering adversary divert SNM with no risk of detection?
    - Level 1.5 -- Which personnel at the facility can defeat the Materials Control & Accounting System?
- Level 2 -- Adequacy. Can a nontampering adversary divert SNM with some level of risk, and does the probability of detecting that adversary meet the NRC criteria?
- Level 3 -- Sensitivity & Collusion. What system states, such as failed components or collusion among employees and adversaries, would allow the adversary to divert SNM? Does the system meet single-failure criteria?
    - Level 3.5 -- Which personnel can cause these failed states?
- Level 4 -- Tampering and Collusion. Can the adversary tamper with the system, through either altering physical systems or colluding with others, in order to divert SNM without detection?

1.1.1 Level 1 - Coverage

The purpose of Level 1 is to determine if a nontampering adversary can divert SNM with no risk of detection. In other words, assuming that no component has failed, is the MC&A system monitoring all potential diversion paths?

To accomplish this purpose, the main analysis in Level 1 requires generating Target Sets (TS), which are lists of elements that the adversary seeking SNM will encounter. A Target Set is defined by an exhaustive enumeration of both the areas and portals used by the adversary in entering and leaving a facility and the process volumes, such as tanks whose state will be altered as the SNM leaves the system. The list of monitors that protect a

Target Set is called a Monitor Target Set (MTS). The data required to define the Monitor Target Sets include a physical description of the plant, data on the field of view of the monitor, data, and information on the adversary.

1.1.2 Level 1.5 - Weak Collusion

The Level 1.5 analysis determines which sets of personnel at the facility can, by their normal (direct) access and/or control, uncover the MTS's of Level 1.

1.1.3 Level 2 - Adequacy

The Level 2 analysis extends the Level 1 analysis to consider the reliability of the system. The system reliability is calculated for each adversary type and for each Monitor Target Set.

We assume that the adversary does not tamper with the system, that he does not have knowledge of the system status except for the operating mode, and that he makes only one attempt to divert SNM. Consequently, the appropriate measure of the system's availability is the probability that a given Monitor Target Set will be uncovered if attacked once by a given adversary type at a random entry time during any given operating mode of the facility.

The calculation of the probability of detection conditioned on adversary type, mode, and monitor Target Set is complicated by the common mode failure problem. Utilities such as electricity or compressed air can fail and thereby cause several MC&A components to fail at once. The utility structure is part of the input to the Level 1 analysis, allowing dependence among components to be modeled explicitly.

1.1.4 Level 3 - Sensitivity

Level 3 analysis includes more sophisticated adversary types with special knowledge of the status of the MC&A system. These adversaries do not tamper with the system, but they do have knowledge of the status of some or all of the MC&A system components.

Complete knowledge is equivalent to observing a status board with a light that goes on for every component that is operational, but goes off for every component that has failed. If Level 3 assumes the adversary has complete knowledge and does not tamper, Level 3 assumes that an adversary will attack only uncovered Target Sets.

1.1.5 Level 3.5 - Strong Collusion

The purpose of Level 3.5 is to determine which sets of facility personnel can cause the MC&A system to fail. This analysis is similar to the analysis of Level 1.5 but includes facility personnel with control of both monitors and the components identified in Level 3.

1.1.6 Level 4 - Tampering Vulnerability

Level 4 asks the question: What are all possible methods to defeat the MC&A system? This is the most sophisticated level of analysis, and if completed successfully and modeled in the same level of detail as the previous levels, includes all of the results of Levels 1 through 3 as special cases. The output from Level 4 is the sets of facility personnel who, by using any means including stealth and deceit, can defeat the MC&A system.

1.2 OVERVIEW OF SAA CODES

As described, the Structured Assessment Approach consists of four major levels and two sublevels. By October, 1979, Levels 1, 1.5, 2, 3, and 4 have been implemented on the Lawrence Livermore National Laboratory OCTOPUS system.

The codes for the levels of the SAA are called CORE codes in this document, since these codes perform the basic, or core, SAA analyses. Two preprocessors prepare data for these codes. The relationship of the codes is shown in Figure 1.2-1. The first of these preprocessors converts the data

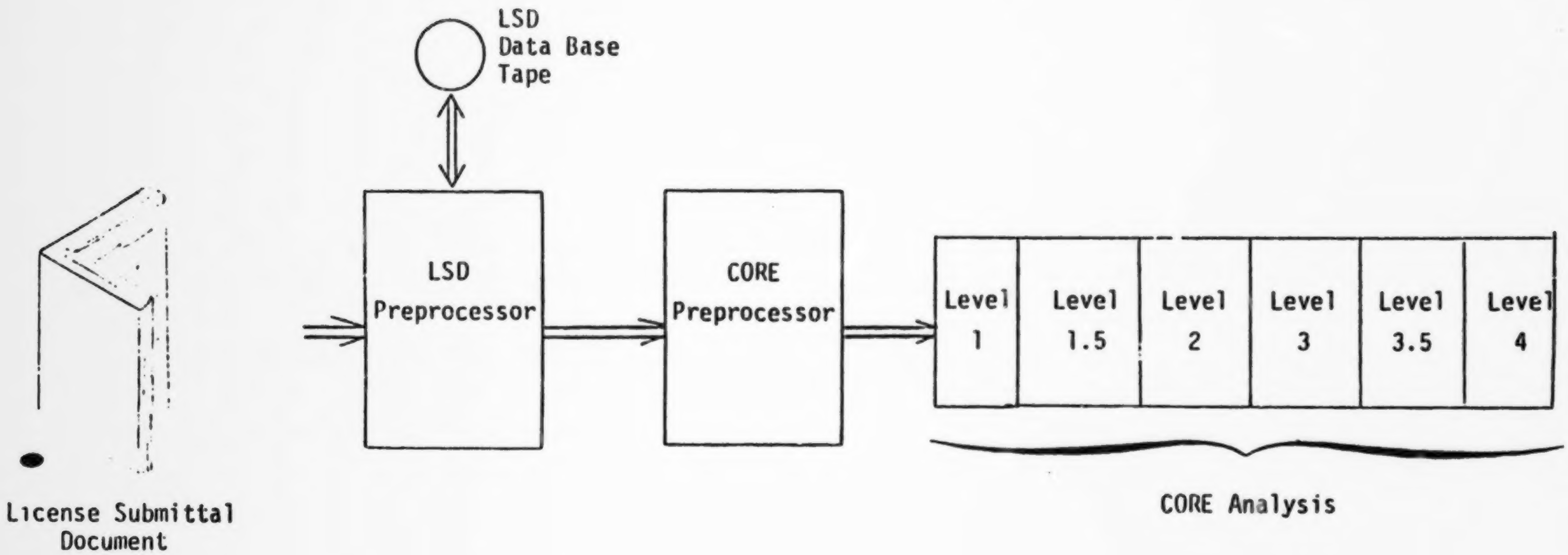

LSD
Data Base
Tape
LSD
Preprocessor
CORE
Preprocessor
Level
1
Level
1.5
Level
2
Level
3
Level
3.5
Level
4
License Submittal
Document
CORE Analysis

FIGURE 1.2-1: Structured Assessment Approach Code Organization

submitted by the license applicant (as per Volume 2 of this report) into a computer-readable data base. This preprocessor, called the LSD preprocessor, also performs the necessary data base searches to construct the input required by the second preprocessor, the CORE preprocessor. Ultimately, the NRC analyst will interact primarily with the LSD processor. When fully implemented, this preprocessor will also perform data consistency.

The CORE preprocessor converts the data base generated by the LSD preprocessor into data arrays required by the Level 1-3.5 codes. The output from the CORE preprocessor consists primarily of bit vectors describing area adjacency, monitor fields of view, process element connections, etc. In addition, the CORE preprocessor generates two types of access and control vectors describing which personnel have the ability to disable components of the MC&A system. The first of these lists, the normal access and control list, is merely the data specified by the applicant in a new format, whereas the second, the extended access and control list, contains SAA-determined additions to this list.

The remainder of this document contains technical descriptions of the basis for each of the codes and code listings with sample outputs.

2.0 STRUCTURED ASSESSMENT APPROACH TECHNICAL DESCRIPTION AND CODES

This section presents the technical description of each level of the SAA analysis codes. As stated above, there are four main levels analysis and two sublevels. The technical descriptions in this section have been generated from internal LLNL technical memoranda written before the codes were developed. Although the technical descriptions may be slightly at variance to the actual code implementation for this reason, the codes are designed to implement the intent of the technical description. Since all of the codes listed in this document are still being developed, they may not be completely debugged. The codes, as listed, have been exercised on the example problem of Volume II and produced the results shown in Volume III. Results shown in Volume III are as they were generated by the SAA Version I computational package as of September 28, 1979, without hand massaging any output data.

Each of the following sections presents a technical description, example output, and code listing for each portion of the SAA. The major blocks of the SAA were shown in Figure 1.2-1. These are:

(1) LSD preprocessor
(2) Core preprocessor
(3) Level 1 - Coverage
(4) Level 1.5 - Weak Collusion Analysis
(5) Level 2/3 - Adequacy
(6) Level 3.5 - Strong Collusion Analysis
(7) Level 4 - Tampering Vulnerability

Level 3.5 has not been implemented as of this date. Implementation is scheduled to begin early in October 1979.

2.1 LICENSE SUBMITTAL DOCUMENT

The data which describes the facility and its Safeguard System to the SAA analysis codes is contained in a document called the License Submittal Document, LSD. This document is described in great detail in Volume II of this report and will not be repeated here.

2.1.1 NRC Control

The SAA is intended to become a production package that an NRC analyst can use on LLNL computers. This control program was developed to act as the interpretor between the user and the SAA codes, and between the SAA codes and the OCTOPUS computer system.

Features:

The software controller of the analysis package is designed to automatically ask the analyst for the data required to run any particular threat and/or facility mode of operation. The controller then issues the necessary commands to the specific SAA programs and provides information and error messages to the user. Currently the preprocessor has been fully encapsulated in the controller program. As individual programs of the SAA are changed, the control program will generally not require any changes. At the conclusion of any turn a record is kept of all instructions the controller issued and their results.

Language:

The controller is written and executed in BCON, a special high—level language developed at LLNL exclusively for the LLNL computer system. The language, although similar in appearance to a limited FORTRAN, is interpreted rather than compiled. The current form of BCON is capable of executing any program in the SAA and all system utilities including executing other controller programs.

2.1.2 Code Listing

The NRC Control Program Code Listing is found on pages 58-66 of the microfiche in the pocket at the end of this document.

2.2 LICENSE SUBMITTAL DOCUMENT PREPROCESSOR

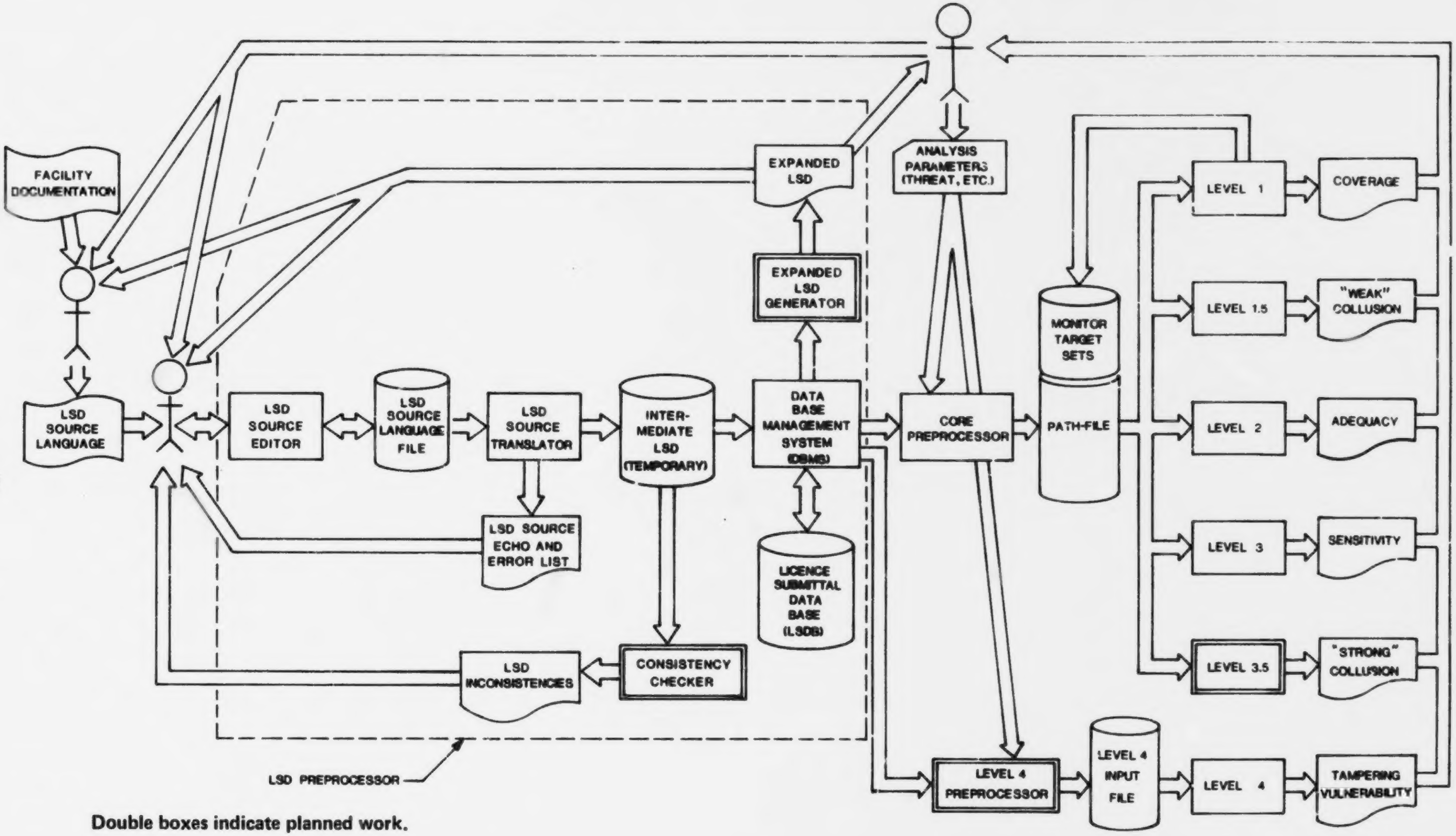
The purpose of the License Submittal Document (LSD) Preprocessor is to convert the data contained in the LSD describing the facility into a form amendable to analysis by computer programs. The LSD preprocessor also finds errors in the input data and prints out the input data in various formats to assist the analyst in verifying the input data.

The flow of data through the SAA is shown in the SAA Information Flow diagram, Figure 2.2-1. First, the NRC license reviewer analyst takes a description of the facility and, following the procedure specified in Volume II of this document, generates a model of the facility.* The format of this model is known as the LSD Source Language and at this stage consists of lines of text on a data coding form. This data is then keypunched into the LSD Preprocessor. Any errors detected in the data are fed back to the operator who resolves the errors and fixes the information in the LSD Source Language and feeds it back to the LSD Preprocessor.

Once the LSD Source Language is error-free, a data base (the License Submital Data Base (LSDB)) is created and the SAA analysis of the facility can proceed. At the analyst's option, an expanded LSD listing is generated explicitly showing various relationships in the LSD data that are only implicit in the LSD Source Language (e.g., a listing of all components that are in a given location, or a list of all monitors covering a particular area or component).

---

*It would be preferable for the NRC to require that Licensee data be submitted in the format used by the SAA.


FACILITY DOCUMENTATION
LSD SOURCE LANGUAGE
LSD SOURCE EDITOR
LSD SOURCE LANGUAGE FILE
LSD SOURCE TRANSLATOR
LSD SOURCE ECHO AND ERROR LIST
INTER-MEDIATE LSD (TEMPORARY)
CONSISTENCY CHECKER
LSD INCONSISTENCIES
DATA BASE MANAGEMENT SYSTEM (DBMS)
EXPANDED LSD GENERATOR
EXPANDED LSD
LICENCE SUBMITTAL DATA BASE (LSDB)
LSD PREPROCESSOR
ANALYSIS PARAMETERS (THREAT, ETC.)
CORE PREPROCESSOR
MONITOR TARGET SETS
PATH-FILE
LEVEL 1
LEVEL 1.5
LEVEL 2
LEVEL 3
LEVEL 3.5
COVERAGE
"WEAK" COLLUSION
ADEQUACY
SENSITIVITY
"STRONG" COLLUSION
LEVEL 4 PREPROCESSOR
LEVEL 4 INPUT FILE
LEVEL 4
TAMPERING VULNERABILITY

Double boxes indicate planned work.

Figure 2.2-1: SAA Information Flow

The Core preprocessor starts that analysis of the SAA by extracting certain portions of the data from the LSDB and passing them to the analysis computer programs (Levels 1-4). Any errors discovered as a result of running the SAA analysis codes are fed back to the NRC analyst, who corrects the errors and resubmits the data to the LSD preprocessor for another cycle of error correction.

2.2.1 Structure of the LSD Preprocessor

The LSD Preprocessor is composed of five computer programs. This section briefly describes the five programs and their interfaces.

The first program is the LSD Source Editor. Referring again to Figure 2.2-1, the LSD Source Language is keypunched into the computer via the program called the LSD Source Editor. The purpose of this program is to generate an exact replica of the LSD Source Language, but in computer-readable format (the LSD Source Language File). Notice the double-ended arrows going into the LSD Source Editor indicating that this is an interactive program. That is, the operator issues commands to the editor and the editor responds to the operator immediately (or nearly so).

The second program is the LSD Source Translator. It accepts the LSD Source Language File created by the LSD Source Editor. The translator reads and prints out the LSD Source File and checks it for a class of errors called syntax errors. These errors are generally typographical (i.e., a letter was keyed in where a number should have been) and are indicated on the LSD printout. If no errors are detected, the LSD Source Translater reformats the data into a form suitable for the next program, the Data Base Management System (DBMS).

Before the DBMS reads the data, however, a program called the Consistency Checker analyzes the data to make sure that it is consistent. Examples of the types of consistency checks performed are: if area 1 is adjacent to area 2, make sure that area 2 is adjacent to area 1 or, if not, that a one-way barrier separates them, or that all signal paths terminate at a response node, or that each monitor has at least one utility and one signal line attached to it. The consistency checker lists any errors it detects and feeds them back to the analyst.

If the consistency checker detects no errors, the data go in to the DBMS which builds and maintains the License Submittal Data Base (LSDB). Any data that the analysis programs need are requested from the DBMS. The DBMS retrieves that data out of the LSDB and passes them to the requesting program.

The fifth program, the Expanded LSD Generator, similarly requests data in various ways from the DBMS, and reformats and prints out the data in form people can read. This printout is known as the Expanded LSD. The Expanded LSD should help the analyst verify the data he has entered. If the analyst detects an error in the Expanded LSD, he can correct these errors at the LSD Source Language, and resubmit the data for another error correction cycle. The status of each of the programs which comprise the LSD preprocessor is given below:

- LSD SOURCE EDITOR: The functions required of the LSD Source Editor are very similar to most interactive text editors. For the Version 1 implementation of the SAA, a text editor known as TRIX, which is available on the OCTOPUS system at LLNL, is used as the LSD Source Editor. Since an editor specifically designed to handle the LSD Source Language would make data entering and correcting errors easier, such an editor should be considered to upgrade the SAA.

- LSD SOURCE TRANSLATOR: This is a new code written in FORTRAN in an as nearly as possible a computer-independent manner. The Version1 code can input and translate a good LSD source file. This means that it can handle LSD Source Files with only minor errors; significant errors may cause the Source Translator to terminate abnormally (read : CRASH). Necessary upgrades to this code include: better error detection and recovery capabilities, better error messages, and bullet-proofing from all possible errors.

- DATA BASE MANAGEMENT SYSTEM: The functional requirements of this system are similar to those of systems that are becoming quite common and are known generically as relational DBMS's. Essentially, a relational DBMS handles any kind of data entity and any kind of relationships between those entities. A relational DBMS builds, accesses, and maintains large data bases (in this case, the License Submittal Data Base). For Version 1 of the SAA, a relational DBMS developed at LLNL, called FRAMIS, is used in this capacity. FRAMIS is currently only available on OCTOPUS, but the FRAMIS implementors are working on a portable version. Several relational DBMS's are commercially available and most of them would be suitable to perform the task FRAMIS is now performing.

- CONSISTENCY CHECKER: This very important piece of the LSD Preprocessor is currently not implemented. We expect the consistency checker to catch many of the errors in the facility input data. If these types of errors are left unchecked, the results coming out of the Levels 1-4 analysis codes could be totally erroneous. For this reason, the implementation of the Consistency Checker must be given a very high priority on the SAA upgrade list.

- EXPANDED LSD GENERATOR: This piece of software is currently not implemented. Tha automatic documentation generation capability of the SAA is represented by this code and thus its usefulness is great. The effort required to develop an Expanded LSD Generator is relatively small, because of the powerful formatting capabilities of FRAMIS (the DBMS feeding the data to the Expanded LSD Generator).

2.2.2 Code Listing

The LSD Processor Program Code Listing is found on pages 1-57 of the microfiche in the pocket at the end of this document.

2.3 CORE PREPROCESSOR

The function of the Core Preprocessor is to convert the LSD data contained in the LSD data base into a form usable by the Level 1 through 3.5

codes. In addition, it condenses the very large physical location networks into smaller networks in order to reduce Level 1 processing time requirements. The Core Preprocessor also expands the authorized control and access lists as specified by the LSD to include possible accesses and controls by each individual at the facility.

2.3.1 Technical Description

This section gives an overview of the data path going from the LSDB to Levels 1-3.5, collectively known as the Core Preprocessor[1]. Figure 2.3-1 presents the Level 1 Preprocessor data flow.

The LSDB, which contains all the MC&A data about a facility required for assessment by the SAA, is maintained by a data base management system called FRAMIS as described in Section 2.2. Data is extracted from the LSDB via a series of requests issued to FRAMIS. Data filling these requests are then passed to the Core Preprocessor.

The Core Preprocessor condenses and reformats this data and outputs it to a file called PATHFILE.

The PATHFILE is then directly used as input by the Level 1, 1.5, 2, 3, and 3.5 analysis codes.

The process in Figure 2.3-1 labeled Core Preprocessor can be further broken down into smaller processes. Figure 2.3-2 is a diagram of that breakdown.

A file containing only that portion of the LSDB which is required for Levels 1-3.5 is first created via FRAMIS. A condensation routine then takes that information, condenses non-monitored areas into super-nodes, and outputs a file called AREAMAP, which maps area numbers into super-node numbers.

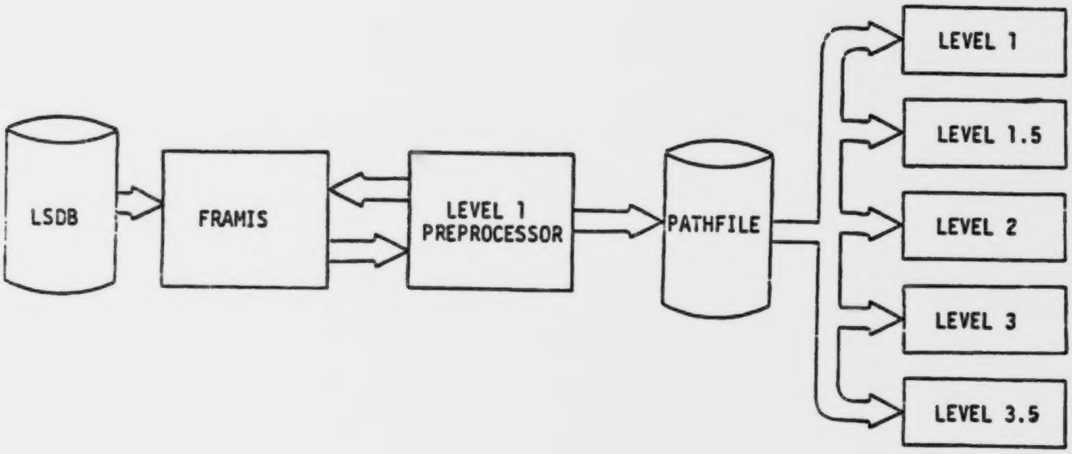

LSDB
FRAMIS
LEVEL 1 PREPROCESSOR
PATHFILE
LEVEL 1
LEVEL 1.5
LEVEL 2
LEVEL 3
LEVEL 3.5

FIGURE 2.3-1: DATA FLOW : LSDB TO LEVEL 1-3.5

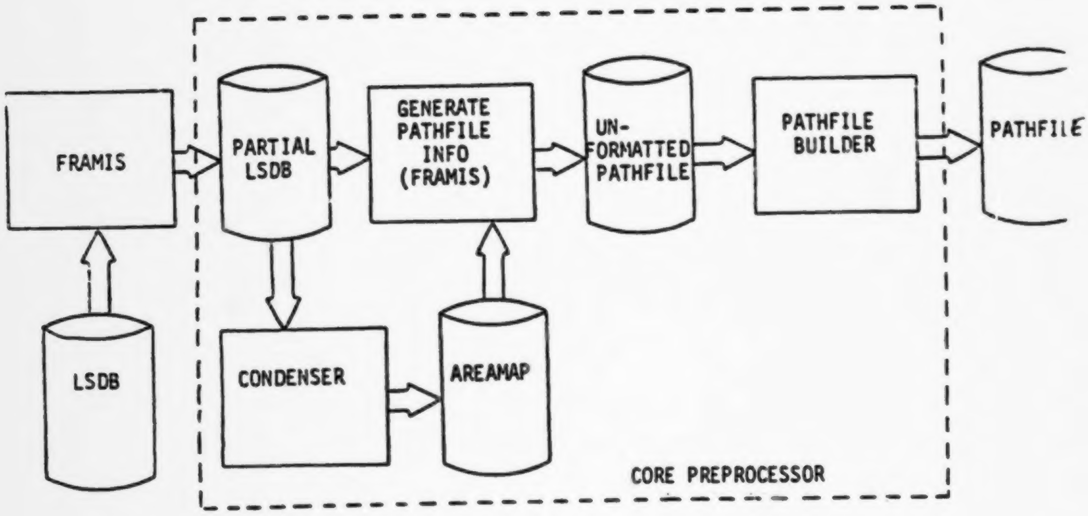

FRAMIS
LSDB
PARTIAL LSDB
GENERATE PATHFILE INFO (FRAMIS)
UN-FORMATTED PATHFILE
PATHFILE BUILDER
PATHFILE
CONDENSER
AREAMAP
CORE PREPROCESSOR

FIGURE 2.3-2: DATA FLOW THROUGH CORE PREPROCESSOR

The AREAMAP is then used to modify the rest of the LSDB data to reflect the results of condensation. The data are then output in a format similar to PATHFILE, but is not packed into bits and bytes as PATHFILE requires. Thus far, all of the operations required for generating the unformatted pathfile are handled by FRAMIS.

Another routine then takes the unformatted pathfile and reformats it into the PATHFILE, which is then fed directly into the Levels 1-3.5 codes.

2.3.2 Condensation

Graphs of facilities may contain a large number of nodes. Processing these graphs to determine all paths from any area to any other area (i.e., reachability matrix) requires very long computational times and a very large memory. An example of a location graph needing excessive computation is shown in Figure 2.3-3 with its associated adjacency matrix in Figure 2.3-4.

The Structured Assessment Approach attempts to minimize computation by four strategems[2]. These are:

(1) Reduce all potential adversary diversion paths (Access, Acquisition, Exit Target Sets) to the paths as seen by the Safeguard System. This means that the SAA maps Target Sets as Monitor Target Sets (MTS).

(2) For any set of MTS's between any node pair, retain only those which are dominant. This means, for example, that if there are 4 MTS's between nodes i and j:

MTS 1 = M1 . M2 . M3
MTS 2 = M1 . M2
MTS 3 = M3 . M4 . M5
MTS 4 = M2 . M3

only MTS 2, MTS 3, and MTS 4 would be retained.

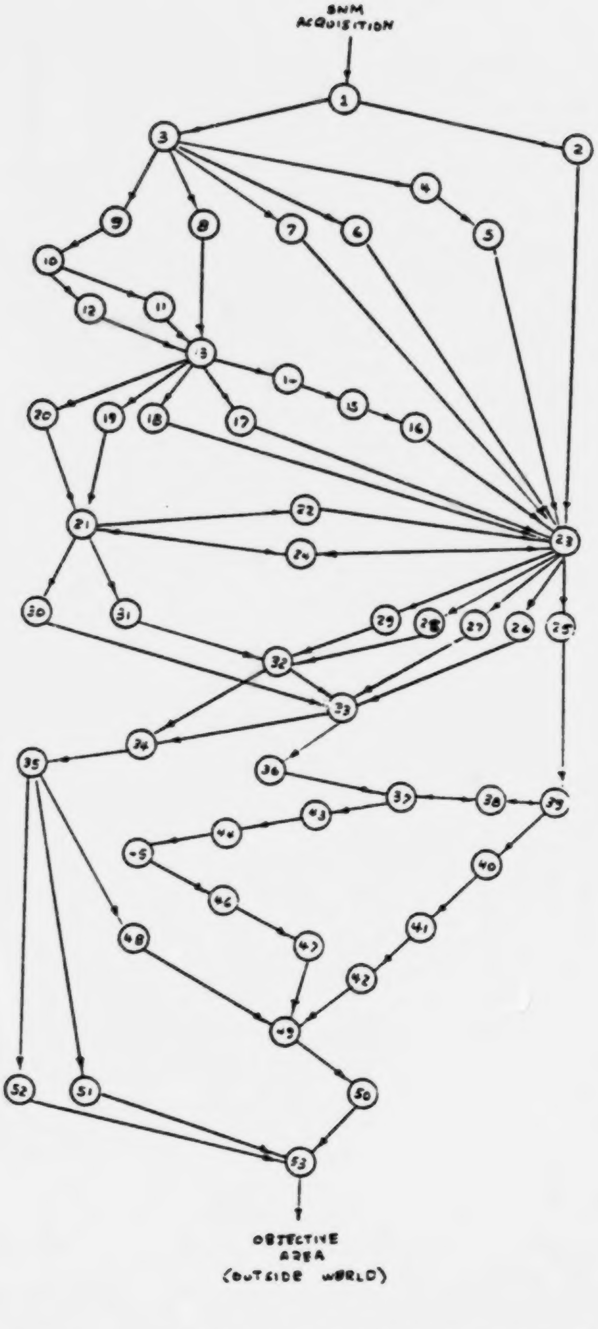

SNM
ACQUISITION
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
OBJECTIVE
AREA
(OUTSIDE WORLD)

FIGURE 2.3-3: Location Graph Which Requires Excessive Computation For Reachability Calculation.

```
1                             LOCATION ADJACENCY MATRIX

                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5
      1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3

 1    0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 2    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 3    0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 4    0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 5    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 6    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 7    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 8    0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 9    0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10    0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11    0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12    0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13    0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14    0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
17    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
18    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
19    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
20    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
21    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
23    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
24    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
25    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
26    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
27    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
28    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
29    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
30    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
31    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
32    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
33    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
34    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
35    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0
36    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
37    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
38    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
39    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
40    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
41    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
42    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
43    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
44    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
45    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
46    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
47    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
48    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
49    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
50    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
51    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
52    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
53    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

FIGURE 2.3-4: Adjacency Matrix For Location Graph of Figure 2.3-4.

(3) In some cases, it is possible to take advantage of the symmetry of the adjacency matrix (for a bidirectional graph) and use just half the number of calculations in the path Reachability algorithm[3].

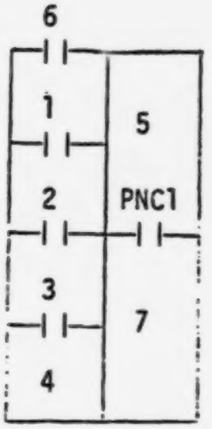
(4) For larger problems that require excessive storage space and computation times, the Core Preprocessor removed all non-essential nodes from the Adjacency Matrix. Nodes that are not required for the problem require both increased computational time and increased storage.

The calculational algorithm now implemented in Level 1 shows the path and the location to monitor mapping calculation in one step. We can also compute more efficiently by using special dominance tests. Using these advantages, we were able to process the Adjacency Matrix shown in Fig. 2.3-4.

The structure of the Warshall reachability algorithm modified for symmetry is:

```
DO 3  K = 1, N
DO 2  I = 1, N-1
DO 1  J = I+1, N
R(I, J) = R(J,I) + R(I,J) + R(I,K) * R(K,J)
1  CONTINUE
2  CONTINUE
3  CONTINUE
```

Areas which are located behind barriers are not reachable from the areas which can be occupied by personnel. A simple example of this situation is shown in Figures 2.3-5a and 2.3-5b.



Physical Layout
(a)

	1	2	3	4	5	6	7
1		1				1	
2	1		1				
3		1		1			
4			1				
5							1
6	1						
7					1		

Binary Adjacency Matrix
(b)

FIGURE 2.3-5: Example of Unreachable Areas

(In this case the barrier symbols normally carried have been set to zero).

Computing the binary reachability for the adjacency matrix of Figure 2.3-5 yields the reachability matrix of Figure 2.3-6.

$$R = \begin{array}{c|c|c|c|c|c|c|c|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline
1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ \hline
2 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline
3 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline
4 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ \hline
5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline
6 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline
7 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline
\end{array}$$

FIGURE 2.3-6: Reachability Matrix For Example of Figure 2.3-5

Areas 5 and 7 are not reachable from the personnel areas. (Actually only reachability from any one node including the outside need be calculated.) Thus, nodes 5 and 7 can be eliminated from the adjacency matrix for Monitor Target Sets for area traversal.

In one special case, the case of an adversary who can penetate barriers, there are no unreachable nodes.

In many cases, many of the areas in the layout will not be monitored. Also, for ease in modeling the facility, the License Submittal Document writer may choose to break up single areas into multiple areas. An example of this is given in the unmonitored hallway of Figure 2.3-7, with the corresponding graph and adjacency matrix in Figures 2.3-8a and 2.3-8b.

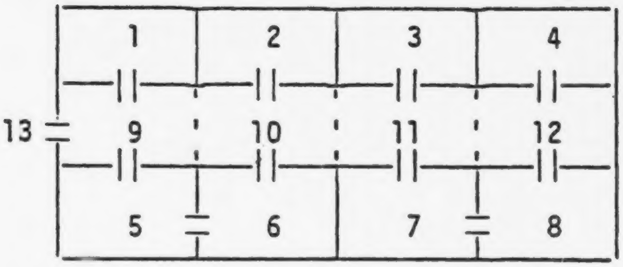
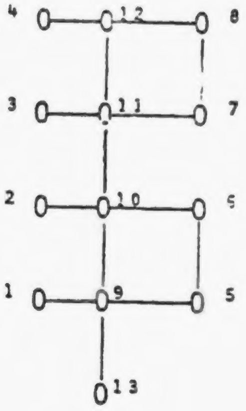


FIGURE 2.3-7: Unmonitored Hallway Example



(a) Graph

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	1	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0	0	1		0	0
7	0	0	0	0	0	0	0	1	0	0	1	0	0
8	0	0	0	0	0	0	1	0	0	0	0	1	0
9	1	0	0	0	1	0	0	0	0	1	0	0	1
10	0	1	0	0	0	1	0	0	1	0	1	0	0
11	0	0	1	0	0	0	1	0	0	1	0	1	0
12	0	0	0	1	0	0	0	1	0	0	1	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	0

(b) Adjacency Matrix

FIGURE 2.3-8: Adjacency Information For Example of Figure 2.3-7

In this case, it seems easier to break up the continuous hallway into separate areas (9 through 12). Marking which nodes are monitored, the graph of Figure 2.3-9 results, where monitored nodes are shown as black.

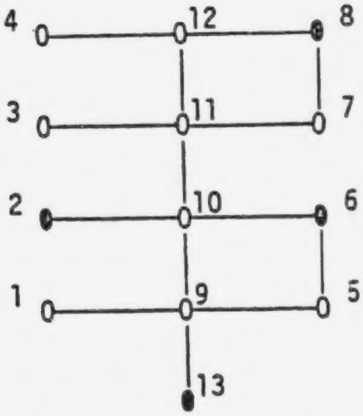
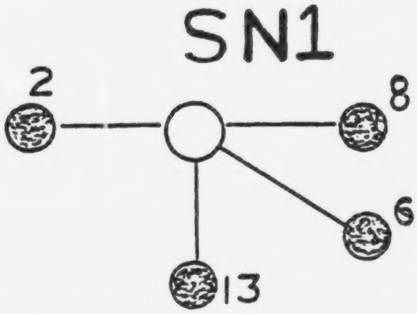


FIGURE 2.3-9: Graph With Monitored Nodes Marked

Since the level 1 SAA code computes Monitor Target Sets, the unmonitored nodes are not important except as connectors between monitored nodes and, of course, as locations of components. Thus, it is desirable to condense all unmonitored nodes into as few super nodes* as possible. For example, the graph in Figure 2.3-9 condenses to the graph in Figure 2.3-10.



SN1 = 1, 5, 3, 7, 11, 12
10, 4, 9

FIGURE 2.3-10: Condensed Graph

---

*These super nodes are analogous to the strong components of directed graph theory.

Obviously, computing the MTS Reachability for the graph of Figure 2.3-10 will be easier than Figure 2.3-9. Figure 2.3-10 also saves significantly in storage.

The algorithm to perform this condensation is straightforward, and is as follows:

1. Construct Adjacency Matrix for Physical Network.
2. Eliminate all Nodes which are monitored.
3. Compute Binary Reachability on the demonitored Matrix.
4. Then, if R(i,j) = 1, node i is equivalent to node j.
5. Store equivalencies (node i = node j).
6. Remove all but one node of each equivalent node set.

An interesting and useful reachability bit of information is the super node equivalents: these indicate uncovered paths between these nodes.

In some cases, the preceeding condensation process will produce graphs of the form shown in Figure 2.3-11. The nodes labeled A, B, and D are fingers and can be removed from the graph before the MTS Reachability calculation.

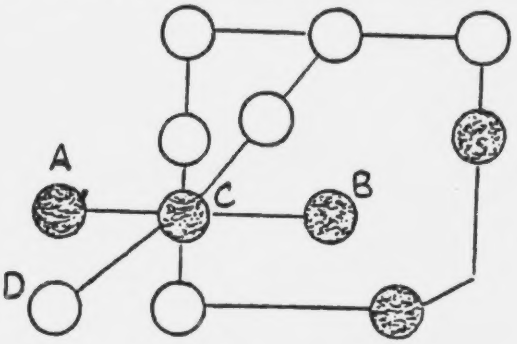


FIGURE 2.3-11: Monitored Graph With Fingers

They can be removed, because for all R(i,j), except for i or j which are the nodes A or B, A and B cannot be part of a dominant Monitor Target Set. This is because MTS's between any (i,j) pair will dominate MTS's which use this singly connected monitored node. Thus, a further reduction algorithm becomes:

$$\text{If for any } j, \sum_{i=1}^{N} R(i,j) = 1$$

Then j is singly connected and cannot be a part of any dominant MTS for any node pair except one involving j.

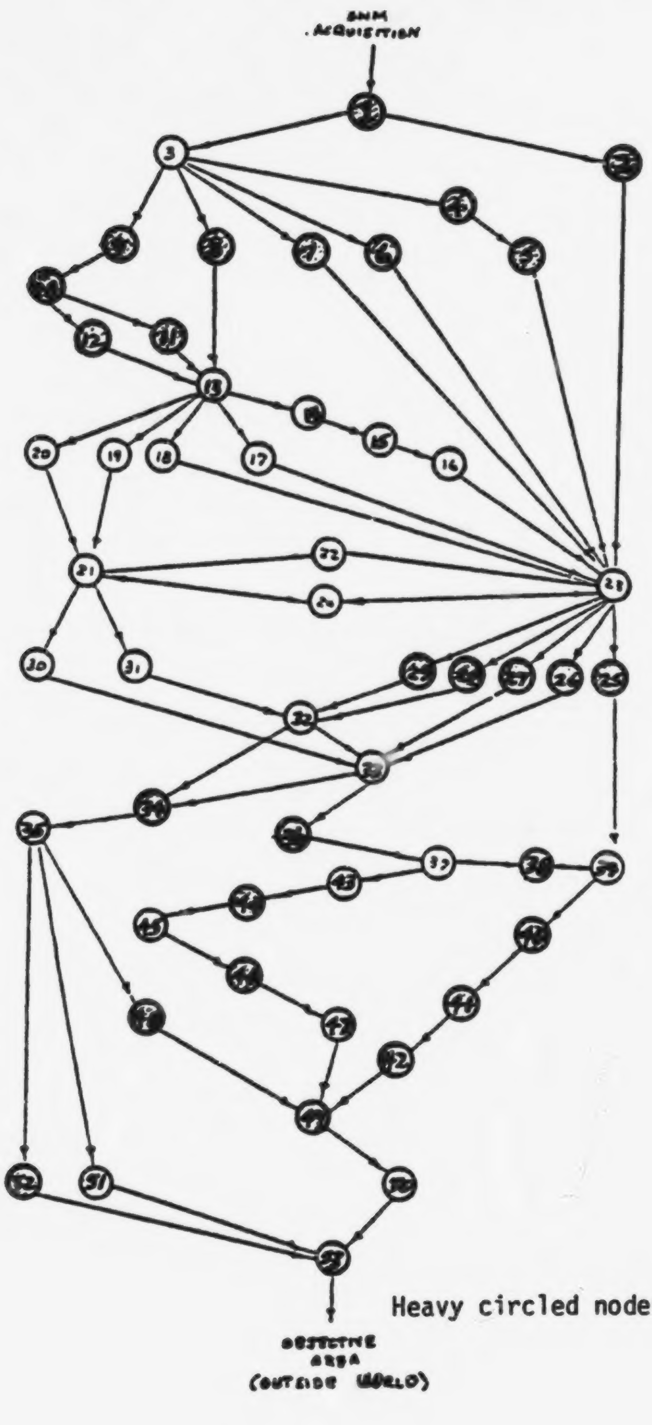
This, potentially, can remove unnecessary nodes from the Reachability calculation.

These nodes will have to be multiplied back in if the eliminated node is required for Target Analysis of Level 1.

Figure 2.3-12 is the Adjacency graph of Figure 2.3-3, where the monitoring nodes (areas in this case) are shaded. The information on the coverage of the monitors would be contained in the field-of—view matrix of the LSD. Removal of the 1's for the monitored nodes yields the Adjacency Matrix which is then processed by a Binary Reachability calculation to yield the Reachability Matrix. The matrices reveal the node equivalencies of Figure 2.3-1.

Super Nodes	Original Nodes
AN 3	3
SN 13	13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 30, 31, 32, 33
SN 35	35
SN 37	37, 43
SN 39	39
SN 47	47, 49, 50, 53
SN 45	45

TABLE 2.3-1: Super Nodes for Fig. 2.3-3.


SNM
ACQUISITION
3
12
13
15
16
17
18
19
20
21
22
23
24
26
25
30
31
32
37
43
42
51
OBJECTIVE
AREA
(OUTSIDE WORLD)

Heavy circled nodes are monitored.

FIGURE 2.3-12: Location Traversal Graph With Monitored Nodes Marked

BEST DOCUMENT AVAILABLE

Thus, the 53 x 53 Adjacency matrix can be reduced to a (28 + 6 =) 34 node matrix.

Now we remove the fingers. Notice in the new graph shown in Figure 2.3-13 that monitored nodes 29, 28, 27, and 26 are connected only to super node 13 and can be a part of a dominant MTS for only themselves. Hence they can be eliminated from the Adjacency to be processed for dominant MTS. Now we have (34 - 4 =) 30 nodes.

2.3.3 Expansion of Access and Control Lists for Facility Personnel

The Structured Assessment Approach codes perform two weak collusion tests prior to full tamper analysis processing on Level 4[4,5]. The tests are called Level 1.5 and Level 3.5*. This section describes an extension to these weak collusion tests which increases the capabilities of these codes to uncover individuals who, if in collusion, could defeat the Material Control System. The Core Preprocessor computes the data for this extension.

The inputs required by the Level 1.5 and Level 3.5 SAA codes are bit vectors which delineate the monitors and/or MC&A system components over which each individual in the facility has control. Typical bit vectors are shown in Figure 2.3-14.

---

*Level 3.5 has not been implemented as of this date.

MODE N

GPAS	M1	M3					. . . .
GPAS/GSAS	M1	M4					. . . .

FIGURE 2.3-14: Typical Control Bit Vectors For Level 1.5 and 3.5

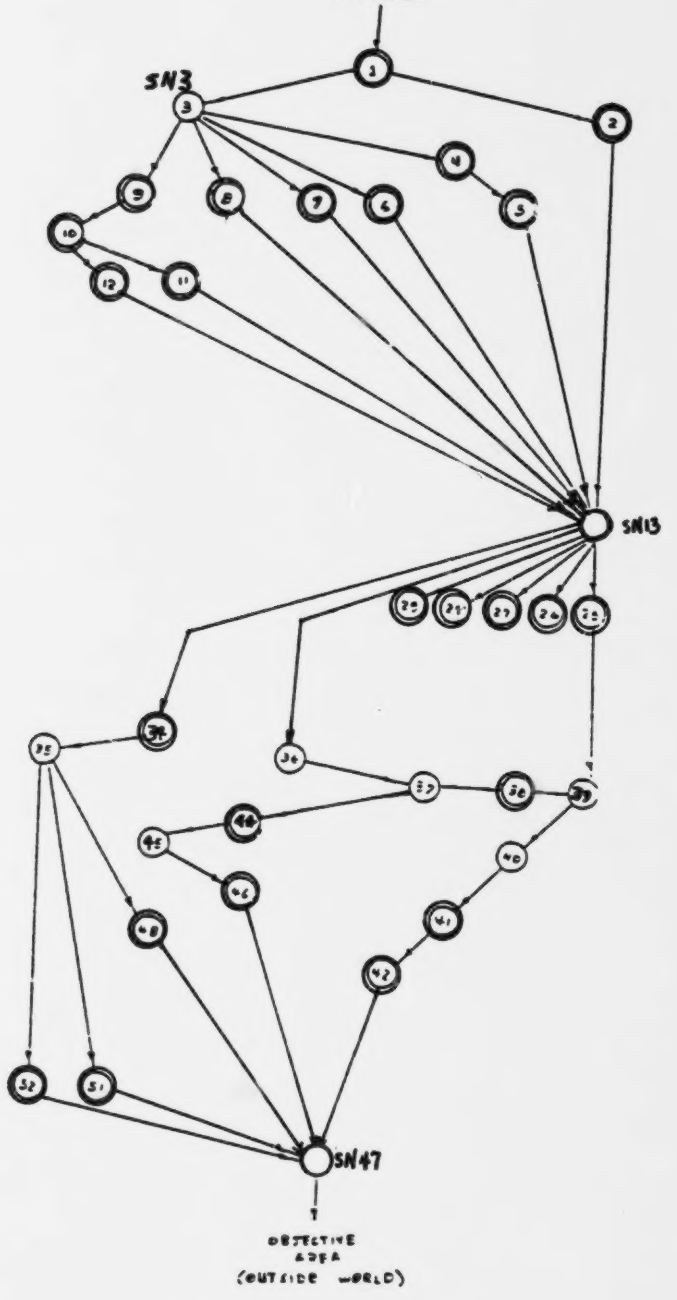

SNM ACQUISITION
SN3
1
2
3
4
5
6
7
8
9
10
11
12
SN13
29
28
27
26
25
34
35
36
37
38
39
40
41
42
44
45
46
48
51
52
SN47
OBJECTIVE AREA (OUTSIDE WORLD)

FIGURE 2.3-13: Condensed Adjacency Graph

The individuals in this case are either actual individuals such as guards, processing operators, supervisors, etc., or combinations of individuals, such as guard primary alarm station/guard secondary alarm station. The combination indicated by a slash (/), referred to as a control vector, means that both individuals (or more) are required to turn off the monitors. There is a set of these control vectors for each defined operating mode (or procedure) at the facility.

These control vectors are generated by a two-step procedure. The first is simply copying the monitors which each individual, or combination of individuals, is authorized to control into the control vector. The second step involves transforming both the locations in which an individual or combination of individuals is authorized to be, and the process elements with which an individual or combination of individuals is authorized to have access to, into the set of monitors which these accesses imply control over. These monitors would be those whose field of view covers authorized locations. For example, if an operator for some mode is authorized to be in location 1 and this location is monitored by monitor 6, then control may be given to the operator over monitor 6, depending on the parameter information about the monitor. Thus, a correlation matrix is used to look up monitors which protect the determined locations. These matrices describe the locations in the field of view of each monitor.

Input to Level 3.5 contains the same monitor control information plus information about utility system control. The utility system control information is determined directly from the authorized control lists of the input to the SAA preprocessor.

Generating the control bit vector as described above yields a description of an individual's or combination of individuals' authorized access and behavior only. The procedure described in this section will expand the control vector to include that which individuals can control even without direct authorization. The procedure will be described for expansion of both the control and access lists.

An individual has access to any area in which he is authorized to be. He also has access to all areas which he can reach from these authorized locations without triggerig a new monitor. Thus, the first expansion of the Level 1.5 and 3.5 access/control lists involves determining where each individual can be outside of his authorized area. This list, as before, must cover all modes of the facility.

To determine the possible locations of all individuals for any given mode at the facility, we would use the condensed location network. This condensation algorithm determines which location nodes are essentially the same due to the fact that they are unmonitored. Figure 2.3-15 gives an example of this situation.

Any individual with access to any of nodes 1,2,3,4,5,6 would have access to all of these nodes. The same is true for nodes 9 and 10. Thus, the authorized access list should be expanded to include any and all possible locations to which any individual could gain access without being detected. For example, the authorized access list of Table 2.3-2a for Figure 2.3-15 is expanded in Table 2.3-2b.

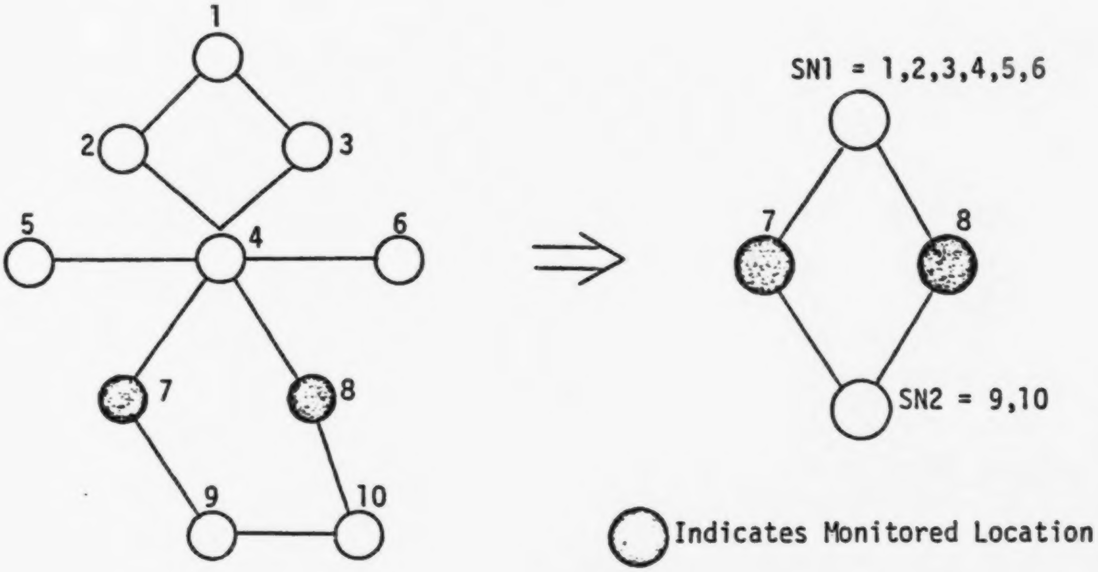


FIGURE 2.3-15: Location Network Condensation

Location	Individual
1	01
2	
3	
4	
5	02
6	
7	03
8	
9	
10	04

(a) Authorized Access List

1=2=3=4=5=6

9=10

Area Equivalences

Location	Condensed Area	Individual
1	SN1	01, 02
2	SN1	01, 02
3	SN1	01, 02
4	SN1	01, 02
5	SN1	02, 01
6	SN1	01, 02
7	7	03
8	8	--
9	SN2	04
10	SN2	04

(b) Expanded Access List

TABLE 2.3-2: First Expansion of Access List.

Figure 2.3-16 presents a flow chart for the first expansion of authorized access. Notice that the location node condensation step should be done first.

The second possible level of access expansion involves determining the locations which are accessible, because an individual has authorized access to an adjacent monitored location.

Assume that an individual is authorized to be in location 7, therefore he could then move into condensed location SN1 or SN2, because all nodes comprising SN1 and SN2 are unmonitored. Thus, the location access list can be expanded to allow individuals with access to monitored locations to move into adjacent unmonitored locations. The result of this step is shown in the expanded access list of Table 2.3-2 for the example of Table 2.3-1 a.

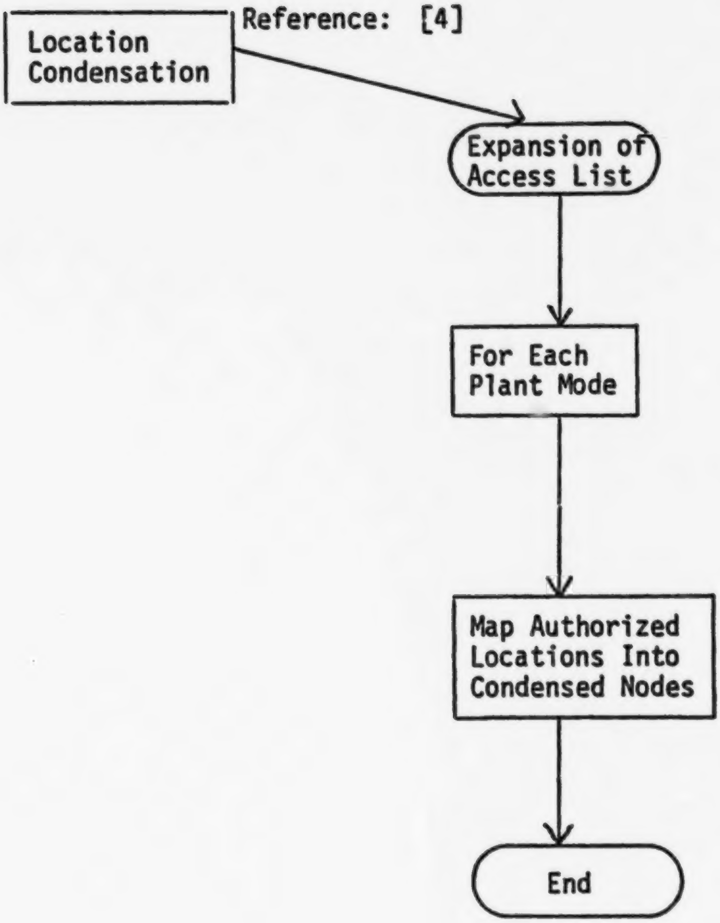

Location Condensation
Reference: [4]
Expansion of Access List
For Each Plant Mode
Map Authorized Locations Into Condensed Nodes
End

FIGURE 2.3-16: First Expansion of Access List

Condensed Location	Original Location	Access Individuals
SN1	1,2,3,4,5,6	01, 02, 03
7	7	03
8	8	
SN2	9,10	07, 03

TABLE 2.3-3: Second Expanded Access List.

We could perform a second level of condensation for each individual for each mode on the condensed location network, and handle the locations in which an individual or combination is authorized to be in, as if they were unmonitored. That is, the condensation procedure is used the same way as before except:

(1) The nodes which are monitored before are now unmonitored if an individual or combination is authorized to be in the location.

(2) This condensation must be done for each mode and each individual or combination within this mode.

The results of this condensation are used to define a new set of locations each individual or combination can reach without being detected. The condensed location network determined by this step must not be used to pass to Level 1 since it applies only to a specific mode. The step described here is shown in the example of Figure 2.3-17, for the condensed network of Figure 2.3-15, where operator 03 has access to node 7.

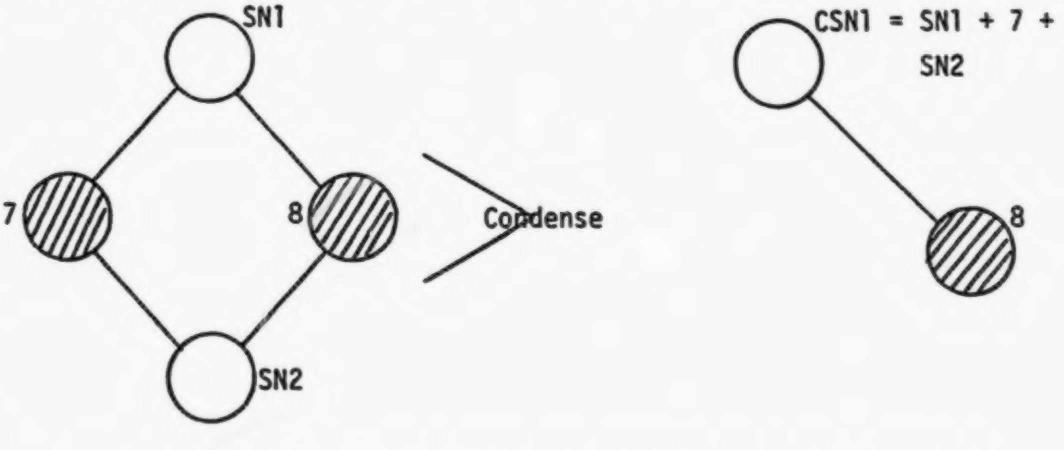


FIGURE 2.3-17: Recondensation For Specific Mode & Individual

This process is shown in the flow chart of Figure 2.3-18. This flow chart accomplishes both steps of the expansion of the access list simultaneously.

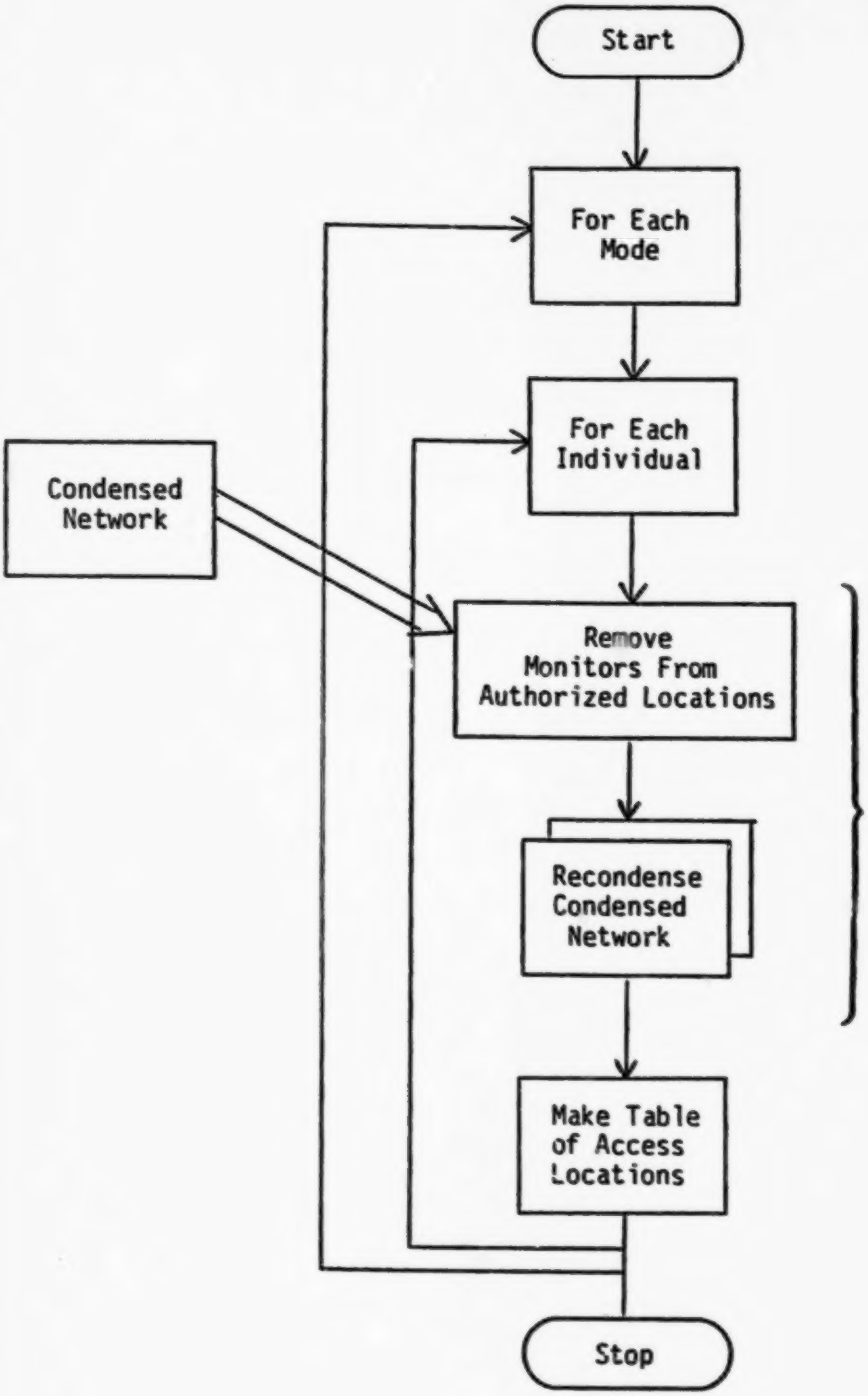
The expanded control vector is created as it was in the previous section. At this point, the monitor control vector will be no different from that which existed before this step, since no new monitored areas have been added to any individual's control list. Control is not expanded until the next step is applied, which can only be applied after this expansion step.

The generation of the control vector in Level 3.5 yields only monitors and other components which an individual is authorized to control via direct control or authorized access. Determining which components can be disabled without detection expands the control list. The assumption which will be made here is: A component which is not tamper protected or monitored is defeatable from the location of that component. Two basic types of tamper monitoring are used: local, such as housing switches; and remote, such as remotely monitored line voltage or signal monitoring. There is, in addition, a third type of tamper monitoring which is a consequence of the second type. Signal transmission components which are used to transmit remote tampering signals (i.e., "I'm OK" signals) are also in effect tamper monitored. Remote tamper monitoring or "supervisory monitoring is used in several actual security systems.

To expand the control lists, we must determine for each mode and each individual or combination the location accesses and the monitors which are not tamper monitored and which possibly could be defeated from these locations. The main difficulty in this procedure is the determination of tamper monitoring.

The MC&A system design may allow for the disabling of certain tamper monitors during certain modes by certain individuals or combinations. This information should be contained in bit vector format as:

01	TM1							


Start
For Each Mode
For Each Individual
Condensed Network
Remove Monitors From Authorized Locations
Recondense Condensed Network
Make Table of Access Locations
Stop

Same process as in Original Network Condensation (Ref. 3)

Table is locations accessable by individual by mode.

FIGURE 2.3-18: Expansion of Access Locations

By using the Tamper Monitor to Monitor (Component) Field-of-View list, we can determine which monitors or components can be defeated without detection, by each tampering individual. Thus, the control bit vector for each individual can be expanded to include this unprotected monitor if two conditions are met:

(1) The individual has access to the monitor location.

(2) The monitor is unprotected by a tamper sensing system, because of either a lack of such a system or the authorized control over the system has been given to the individual.

This procedure is shown in the flow chart of Figure 2.3-19.

We will now discuss the generation of the component Tamper Protection List. Determining whether a component is tamper monitored is straight forward if it is locally monitored as considered earlier. If the device is remotely monitored, components which are between it and the remote monitor will, in general, also be tamper protected. Thus, to determine all tamper-protected components it is necessary to find the path between the component and the tamper monitor. This path should contain all transmission and utility supplies. Its generation is identical to a Level 2 path generation now used. Thus, we must generate a path for each unmonitored component. If this path contains two or more terms, only those elements of the path which are common are tamper protected. An example of both cases is shown in Figure 2.3-20.

The flow chart for the generation of the Component Tamper Protection List is given in Figure 2.3-21. Notice that this list is independent of "individuals" control. This operation must be done before the procedure described in the flow chart of Figure 2.3-19.

2.3.4 Code Listing

The code listing is found on pages 67-98 of the microfiche in the pocket at the end of this document.

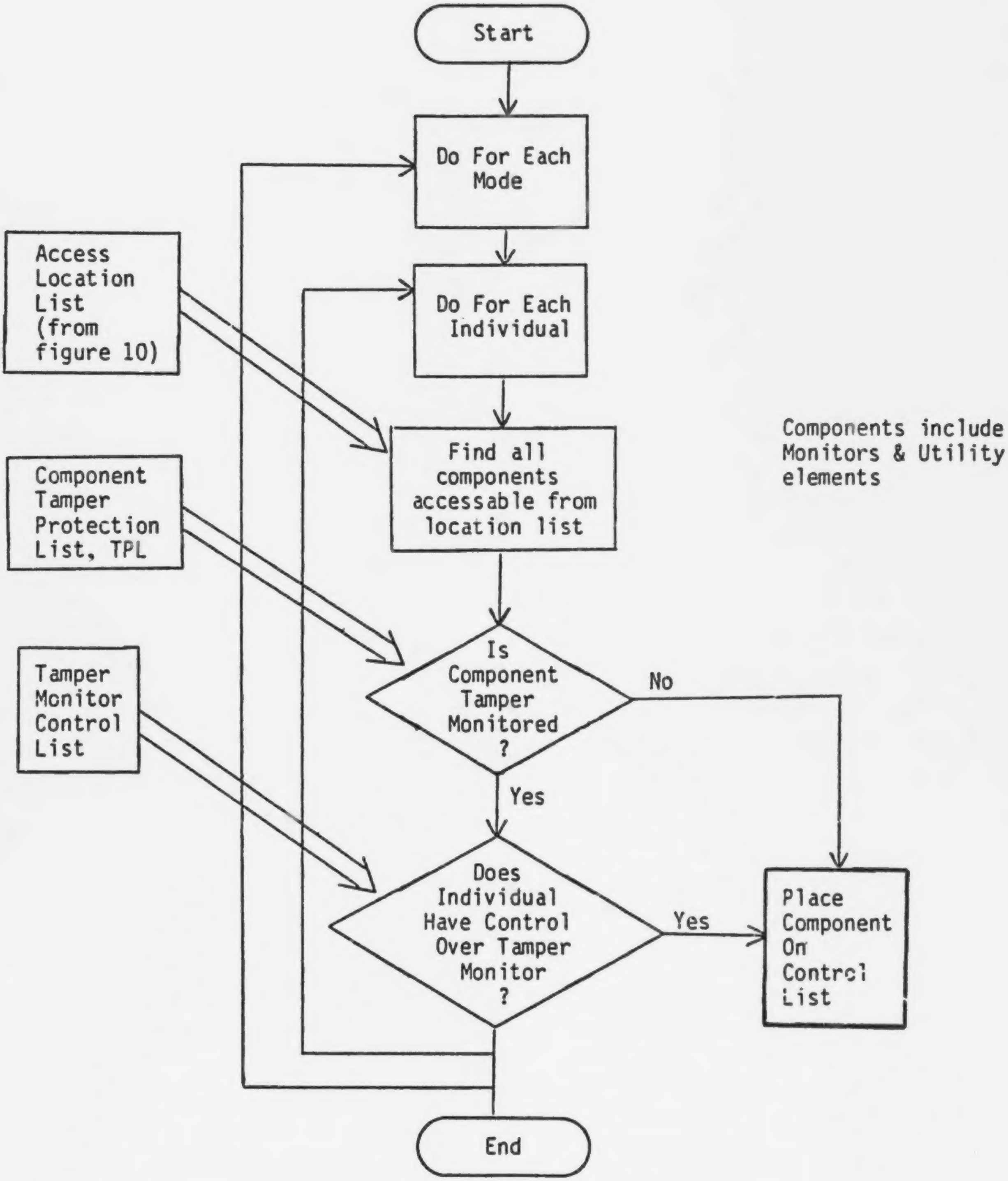
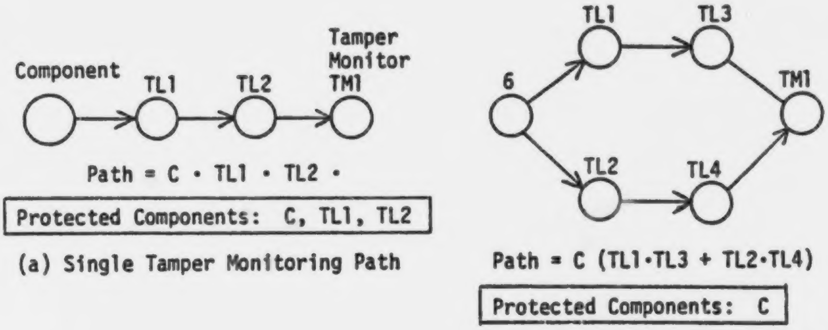

Start
Do For Each Mode
Access Location List (from figure 10)
Do For Each Individual
Find all components accessable from location list
Components include Monitors & Utility elements
Component Tamper Protection List, TPL
Is Component Tamper Monitored ?
No
Yes
Tamper Monitor Control List
Does Individual Have Control Over Tamper Monitor ?
Yes
Place Component On Control List
End

FIGURE 2.3-19: Expansion of Individual Control List


Component
TL1
TL2
Tamper Monitor TM1
Path = C · TL1 · TL2 ·
Protected Components: C, TL1, TL2
6
TL1
TL3
TM1
TL2
TL4
Path = C (TL1·TL3 + TL2·TL4)
Protected Components: C

(a) Single Tamper Monitoring Path

(b) Dual Tamper Monitoring Paths

FIGURE 2.3-20: Indirect Tamper Protection

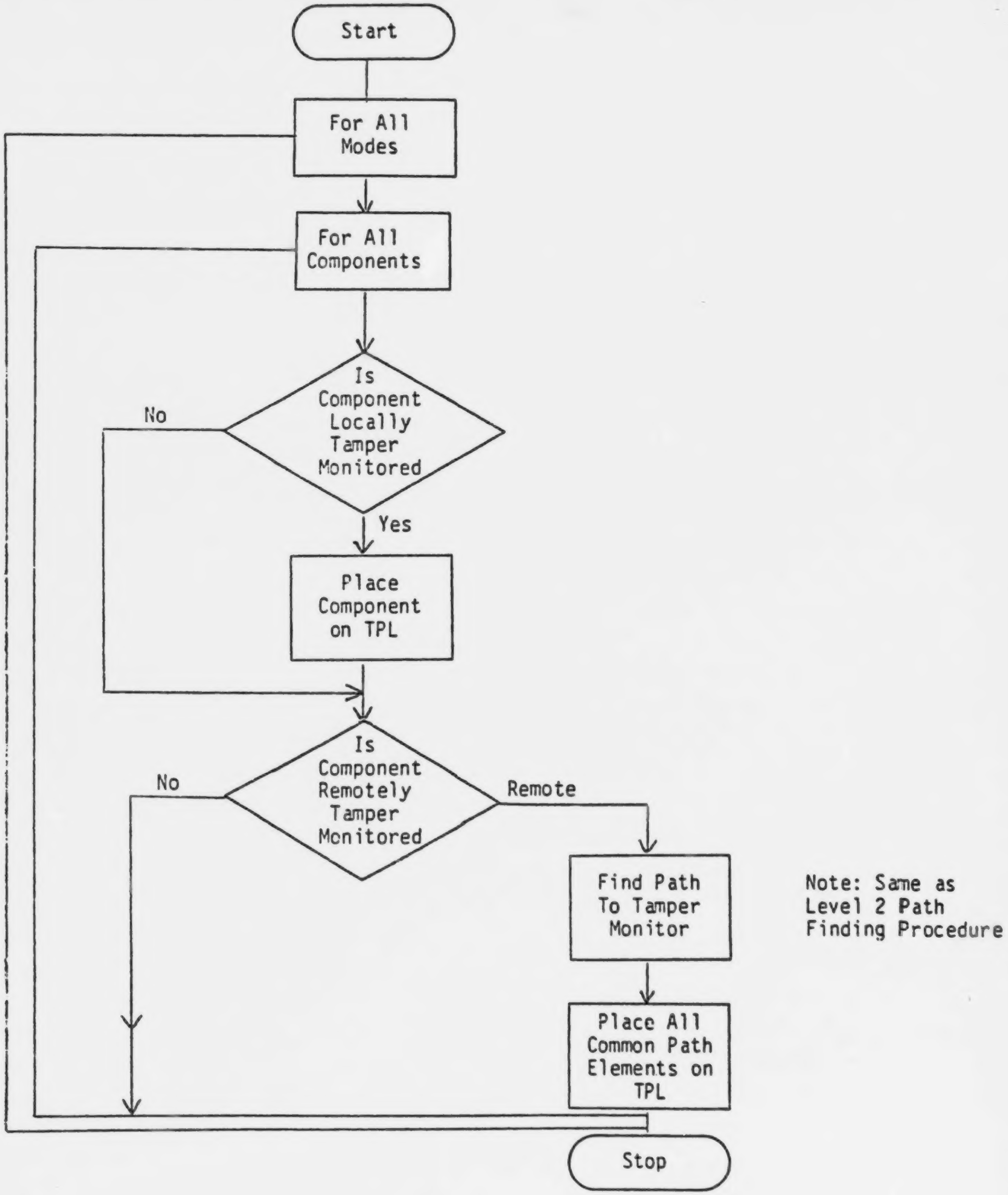

Start
For All Modes
For All Components
Is Component Locally Tamper Monitored
No
Yes
Place Component on TPL
Is Component Remotely Tamper Monitored
No
Remote
Find Path To Tamper Monitor
Note: Same as Level 2 Path Finding Procedure
Place All Common Path Elements on TPL
Stop

FIGURE 2.3-21: Determination of Tamper Protection List

2.4 LEVEL 1 - MONITOR TARGET SET COVERAGE

The purpose of Level 1 of the Structured Assessment Approach[6] is to ensure that the MC&A System Monitor covers all possible diversion paths. It determines all these possible diversion paths in terms of unique monitored sets. Leaving a diversion path unmonitored, where any step of an adversary's diversion sequence is not covered, makes it possible for a non-tampering adversary to divert material without detection. Unmonitored diversion paths are not found in a well—designed facility.

2.4.1 Technical Description

The main activity in Level 1 is to generate target sets (TS), which are lists of elements that will be encountered by an adversary seeking material. A target set is an exhaustive enumeration of the areas and portals used by the adversary in entering and leaving a facility, and the process volumes, such as tanks, whose state will be altered as the SNM leaves the system. The list of monitors that protect a target set is called a monitor Target Set (MTS). The data required to define the Monitor Target Sets include a physical description of the plant, the data on the field of view of the monitors, and information on the adversary. The calculations of the coverage of the Target Sets are done from the point of view of the Safeguard System, that is, Monitor Target Sets are computed directly. An example of the results of computing these Monitor Target Sets is shown in Figure 2.4-1. In this figure, each row is a unique diversion path as seen by the Safeguard System. The columns indicate the monitors which cover each path.

Diversion scenarios are broken into three distinct phases, Acquisition, Exit, and Entry. The Acquisition Target Set (ATS) is all monitors which cover the removal of material from the chemical or physical processing. For example, if an adversary were to remove liquid from a storage tank and if the liquid level were monitored, the Level 1 code would determine that there was one MTS for acquisition. This set would contain the liquid level monitor.

```
1                               MONITOR TARGET SET VS. MONITOR MATRIX
  MTS                           MONITOR INTEGER LABEL
                               1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
             1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2

    1        1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● 1 ● ● 1
    2        1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● 1 ● 1
    3        1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 1
    4        1 ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● 1 ● ● 1
    5        1 ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● 1 ● 1
    6        1 ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 1
    7        1 ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● 1 ● ● 1
    8        1 ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● 1 ● 1
    9        1 ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 1
   10        1 ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● 1 ● ● 1
   11        1 ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● 1 ● 1
   12        1 ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 1
   13        1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● 1 1 1 1 ● ● ● 1
   14        1 ● 1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● 1 ● ● 1
   15        1 ● 1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● 1 ● 1
   16        1 ● 1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 1
   17        1 ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● 1 1 1 1 ● ● ● 1
   18        1 ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● 1 1 1 1 ● ● ● 1
   19        1 ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● 1 1 1 1 ● ● ● 1
   20        1 1 ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● 1 ● ● ● 1 1 1 1 ● ● ● 1
   21        1 1 ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● 1 1 1 ● ● ● ● ● ● ● 1
   22        1 ● 1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 ● ● ● ● 1 1 1 1 ● ● ● 1
   23        1 ● ● ● 1 ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● 1 ● ● ● 1 1 1 1 ● ● ● 1
   24        1 ● ● ● 1 ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● 1 1 1 ● ● ● ● ● ● ● 1
   25        1 ● ● ● ● 1 ● ● ● ● ● ● ● 1 ● ● ● ● ● ● 1 ● ● ● 1 1 1 1 ● ● ● 1
   26        1 ● ● ● ● 1 ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● 1 1 1 ● ● ● ● ● ● ● 1
   27        1 ● ● ● ● ● 1 ● ● ● ● ● ● 1 ● ● ● ● ● ● 1 ● ● ● 1 1 1 1 ● ● ● 1
   28        1 ● ● ● ● ● 1 ● ● ● ● ● ● 1 ● ● ● ● ● ● ● 1 1 1 ● ● ● ● ● ● ● 1
   29        1 ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● 1 ● ● 1
   30        1 ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● 1 ● 1
   31        1 ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 1
   32        1 ● ● ● ● ● ● 1 1 1 1 ● 1 ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● 1 ● ● 1
   33        1 ● ● ● ● ● ● 1 1 1 1 ● 1 ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● 1 ● 1
   34        1 ● ● ● ● ● ● 1 1 1 1 ● 1 ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● 1 1
   35        1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1
   36        1 ● 1 1 ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● 1 ● ● ● 1 1 1 1 ● ● ● 1
   37        1 ● 1 1 ● ● ● ● ● ● ● ● ● 1 ● ● ● ● ● ● ● 1 1 1 ● ● ● ● ● ● ● 1
   38        1 ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1
   39        1 ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1
   40        1 ● ● ● ● ● 1 ● ● ● ● ● ● ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1
   41        1 ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1 ● ● ● ● 1 1 1 1 ● ● ● 1
   42        1 ● ● ● ● ● ● 1 1 1 1 ● 1 ● ● ● ● ● ● 1 ● ● ● ● 1 1 1 1 ● ● ● 1
   43        1 ● 1 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1
   44        1 ● ● ● ● ● ● 1 1 1 1 1 ● 1 ● ● ● ● ● ● 1 ● ● ● 1 1 1 1 ● ● ● 1
   45        1 ● ● ● ● ● ● 1 1 1 1 1 ● 1 ● ● ● ● ● ● ● 1 1 1 ● ● ● ● ● ● ● 1
   46        1 ● ● ● ● ● ● 1 1 1 1 ● 1 1 ● ● ● ● ● ● 1 ● ● ● 1 1 1 1 ● ● ● 1
   47        1 ● ● ● ● ● ● 1 1 1 1 ● 1 1 ● ● ● ● ● ● ● 1 1 1 ● ● ● ● ● ● ● 1
   48        1 ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1
   49        1 ● ● ● ● ● ● 1 1 1 1 ● 1 ● ● ● ● ● ● 1 1 1 1 1 ● ● ● ● ● ● ● 1
```

```
     DOMINANT MONITOR TARGET SETS (MTS) IN LIST FORM

(MTS)   MONITOR INTEGER LABELS

(  1)   1   2  19  29  32
(  2)   1   2  19  30  32
(  3)   1   2  19  31  32
(  4)   1   5  19  29  32
(  5)   1   5  19  30  32
(  6)   1   5  19  31  32
(  7)   1   6  19  29  32
(  8)   1   6  19  30  32
(  9)   1   6  19  31  32
( 10)   1   7  19  29  32
( 11)   1   7  19  30  32
( 12)   1   7  19  31  32
( 13)   1   2  20  25  26  27  28  32
( 14)   1   3   4  19  29  32
( 15)   1   3   4  19  30  32
( 16)   1   3   4  19  31  32
( 17)   1   5  20  25  26  27  28  32
( 18)   1   6  20  25  26  27  28  32
( 19)   1   7  20  25  26  27  28  32
( 20)   1   2  14  21  25  26  27  28  32
( 21)   1   2  14  22  23  24  32
( 22)   1   3   4  20  25  26  27  28  32
( 23)   1   5  14  21  25  26  27  28  32
( 24)   1   5  14  22  23  24  32
( 25)   1   6  14  21  25  26  27  28  32
( 26)   1   6  14  22  23  24  32
( 27)   1   7  14  21  25  26  27  28  32
( 28)   1   7  14  22  23  24  32
( 29)   1   8   9  10  11  12  19  29  32
( 30)   1   8   9  10  11  12  19  30  32
( 31)   1   8   9  10  11  12  19  31  32
( 32)   1   8   9  10  11  13  19  29  32
( 33)   1   8   9  10  11  13  19  30  32
( 34)   1   8   9  10  11  13  19  31  32
( 35)   1   2  20  21  22  23  24  32
( 36)   1   3   4  14  21  25  26  27  28  32
( 37)   1   3   4  14  22  23  24  32
( 38)   1   5  20  21  22  23  24  32
( 39)   1   6  20  21  22  23  24  32
( 40)   1   7  20  21  22  23  24  32
( 41)   1   8   9  10  11  12  20  25  26  27  28  32
( 42)   1   8   9  10  11  13  20  25  26  27  28  32
( 43)   1   3   4  20  21  22  23  24  32
( 44)   1   8   9  10  11  12  14  21  25  26  27  28  32
( 45)   1   8   9  10  11  12  14  22  23  24  32
( 46)   1   8   9  10  11  13  14  21  25  26  27  28  32
( 47)   1   8   9  10  11  13  14  22  23  24  32
( 48)   1   8   9  10  11  12  20  21  22  23  24  32
( 49)   1   8   9  10  11  13  20  21  22  23  24  32
```

```
                          MONITOR LABELS

MONITOR   UNIT    DESCRIPTION      FIELD OF VIEW  LOCATION
INTEGER   MODEL                      INTEGER       SYMBOL
 LABEL    LABEL                       LABEL         LABEL

   1        A     ULTRASONIC            1            A16
   2        C     BALANCE SWITCH        2            F29
   3        B     MICROWAVE             4            A14
   4        C     BALANCE SWITCH        5            F28
   5        C     BALANCE SWITCH        6            F30
   6        C     BALANCE SWITCH        7            F31
   7        C     BALANCE SWITCH        8            F43
   8        C     BALANCE SWITCH        9            F42
   9        G     GUARD VISUAL         10            A13A
  10        D     EMN RADIATION        10            A13A
  11        E     METAL                10            A13A
  12        C     BALANCE SWITCH       11            F40
  13        C     BALANCE SWITCH       12            F42A
  14        C     BALANCE SWITCH       25            F20
  15        C     BALANCE SWITCH       26            F22
  16        C     BALANCE SWITCH       27            F23
  17        C     BALANCE SWITCH       28            F10
  18        C     BALANCE SWITCH       29            F11
  19        F     FENCE IR             34            F1A
  20        C     BALANCE SWITCH       36            F21
  21        C     BALANCE SWITCH       38            F16
  22        C     BALANCE SWITCH       40            F7
  23        B     MICROWAVE            41            A20
  24        C     BALANCE SWITCH       42            F6
  25        B     MICROWAVE            44            A8
  26        G     GUARD VISUAL         44            A4
  27        D     EMN RADIATION        44            A4
  28        E     METAL                44            A4
  29        C     BALANCE SWITCH       48            F5
  30        C     BALANCE SWITCH       51            F2
  31        C     BALANCE SWITCH       52            F3
  32              ACCOUNTING SYS.       1            A16
```

FIGURE 2.4-1: Monitor Target Sets

The Exit Target Set (EXTS) is all monitors which cover the physical path that an adversary would have to use to move from the material acquisition point to the outside of the facility. There may be many EXTS's for each Acquisition Target Set.

The Entry Target Set (ENTS) is all monitors which cover the physical path that an adversary would have to use to get to the material acquisition point. There may be many ENTS's for each Acquisition Target Set.

The basic analysis technique used in Level 1 of the structured assessment of an MC&A System is shown in Figure 2.4-2. The first step is identifying all the potential access points where the adversary could obtain SNM. The NRC Material Control license reviewer then selects those access points he wishes to consider. The second step is identifying all material movement paths to these access points from any source of this material. These lead to the ATS. The third step is identifying all adversary paths through the facility to and from the material access points. These lead to the EXTS's and ENTS's.

Typical ATS, EXTS's, and ENTS's are shown in Figure 2.4-3 and in the Level 1 sample output (Section 2.4.2). Figure 2.4-3 also gives a typical link list which shows the information on which EXTS's and ENTS's connect to which ATS's. It should be noted that all Target Sets are in reality Monitor Target Sets, that is, they are specified from the point of view of the Safeguard system. The coding Level 1 is broken into 4 main sections: area reachability, process element target selector, monitor target set generator, and an output summarizer. The following sections describe each of these steps:

2.4.1.1 Area Reachability. The area reachability section reads the area adjacency and the area monitor field-of-view information from the input file and combines that information into an Adjacency Matrix.

The adjacency array is a square array showing adjacent and monitored areas. For example, an area diagram with six nodes and three monitors would have the adjacency matrix of Figure 2.4-4.*

---

*In the SAA Level 2 code, monitors are actually represented in a bit vector format.

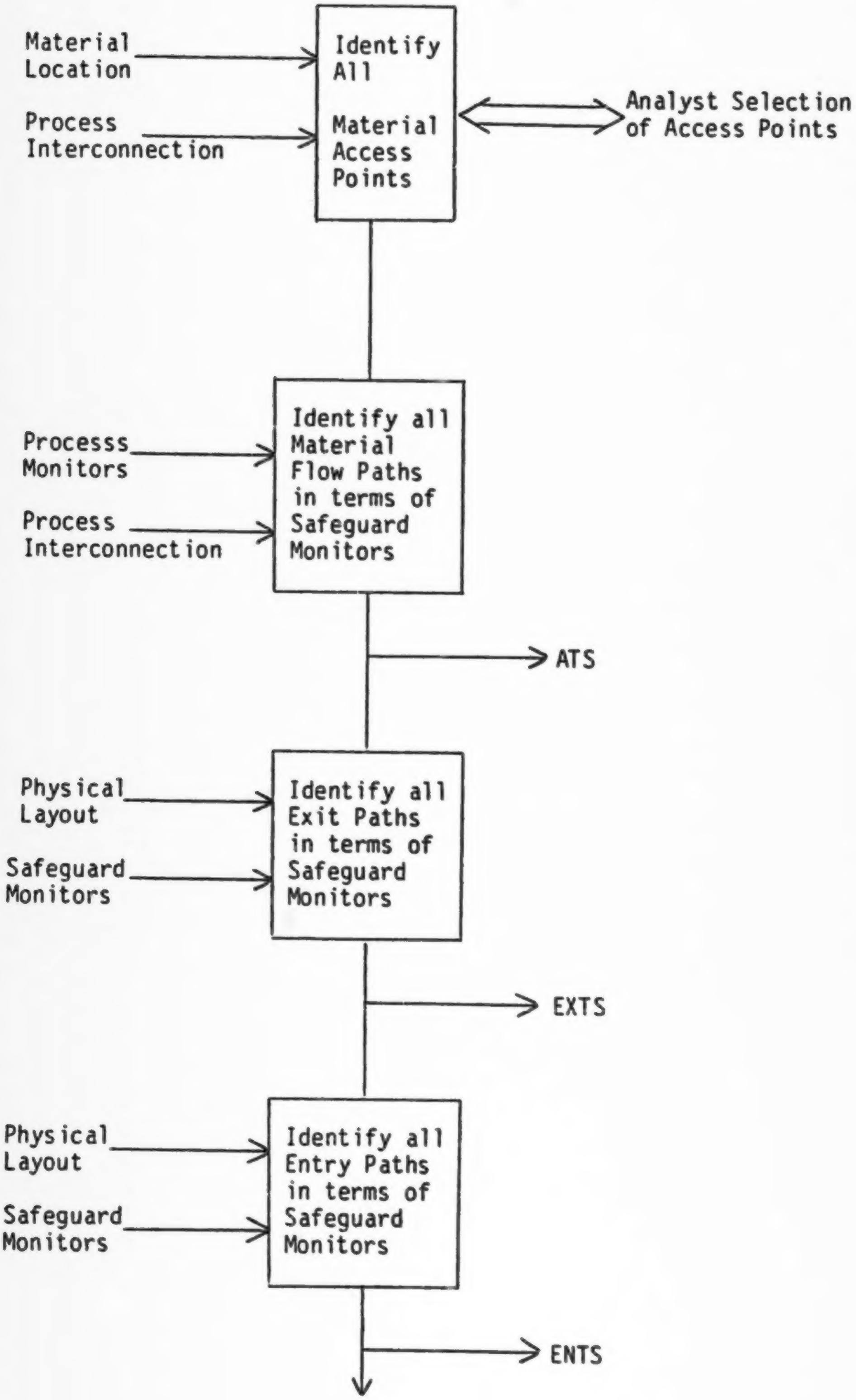

Material Location
Process Interconnection
Identify All Material Access Points
Analyst Selection of Access Points
Processs Monitors
Process Interconnection
Identify all Material Flow Paths in terms of Safeguard Monitors
ATS
Physical Layout
Safeguard Monitors
Identify all Exit Paths in terms of Safeguard Monitors
EXTS
Physical Layout
Safeguard Monitors
Identify all Entry Paths in terms of Safeguard Monitors
ENTS

FIGURE 2.4-2: Conceptual Level 1 Flow

Aquisition Target Sets	B1, Level Monitor	LS1, Valve Limit Switch	LS2, Valve Limit Switch	LS3, Valve Limit Switch	
1	1	1	0	0	
0	0	0	0	0	Unprotected set
2	1	1	0	1	
3	0	0	0	1	

(a) Typical Acquisition Target Sets

Exit Target Set	G1, Guard 1	MW1, Microwave Intrusion Detection	G2, Guard 2
0	0	0	0
1	1	1	0
2	0	0	1

(b) Typical Exit Target Sets

Entry Target Set	G1	MW1	G2
0	0	0	0
1	1	1	0
2	0	0	1

(c) Typical Entry Target

ATS #	EXTS #	ENTS #
0	0	0
1	0	2
2	1,2	1,2
3		

(d) Typical ATS, EXTS, ENTS, Link List

FIGURE 2.4-3: Typical Monitor Target Sets

	1	2	3	4	5	6
1	0	M1	M2	1	0	0
2	M1	0	0	0	M1· M3	0
3	M2	0	0	0	M2· M3	0
4	1	0	0	0	0	0
5	0	M1· M3	M2· M3	0	0	M3
6	0	0	0	0	M3	

FIGURE 2.4-4: Adjacency Matrix

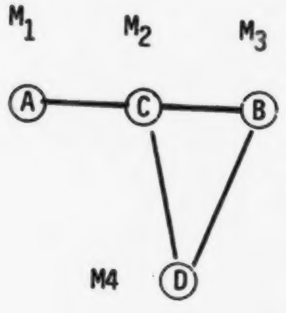
In this adjacency matrix, a 1 indicates physical adjacency with no monitors, and a symbol indicates to pass from area i to area j will activate monitor m. Warshall's reachability algorithm[7] is then used to calculate reachability between all areas in terms of the total set of the least monitors which need to be passed to connect any of these areas.

```
Warshall's Algorithm:
     R = reachability array
     A = adjacency array
Start with R = A
     DO *** K = 1,N
     DO *** I = 1,N
     DO *** J = 1,N
     R(I,J) = R(I,J)+R(I,K)*R(K,J)

     *** CONTINUE
     where N is the number of areas.
```

Dominance testing is done as the calculations in Warshall's algorithm are performed. This is done to eliminate duplicate and non-minimum paths. To avoid excess processing some paths are flagged as being dominant by definition; these include diagonals, original adjacencies, and unmonitored paths. Each path (reachability element) will then be a unique path from the Safeguard System point of view.

An example of the result of dominance testing is shown in Figure 2.4-5.



M_1 covers Area A

M_2 covers Area C

M_3 covers Area B

M_4 covers Area D

(a) Monitored Area Graph

$$R(A,B) = M_1 M_2 M_3 + M_1 M_2 M_3 M_4.$$

(b) Paths between A and B.

$$R(A,B) = M_1 \quad M_2 \quad M_3$$

(c) Dominant Path

FIGURE 2.4-5: Dominance Testing

Any open path, i.e., one with no monitors, is dominant over all others.

2.4.1.2 Target Selection. The process element target selection routine uses the process element and material information from the LSD as well as the reachability data from the area reachability calculation.

Each process element of type 1 is considered a source of material. The flow path from that source is calculated. Any element which (1) is reachable from the outside regardless of monitor coverage, (2) has flow if a vacuum is applied at that point, and (3) is an attractive type, is considered as a target.

All targets are listed with pertinent data, and the NRC reviewer is given an opportunity to select among these for further SAA analysis. Any element which contains a non-free draining volume is considered a secondary source, and is processed as potential source of target material.

The NRC reviewer-selected targets, along with the necessary valves and pumps needed to allow material flow, the operational mode, and process monitor coverage information, are then passed to the Acquisition Target Set Generator.

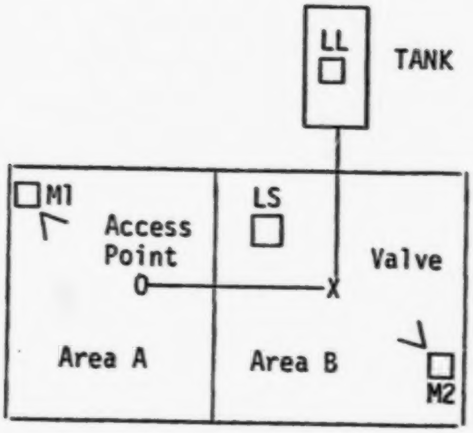
2.4.1.3 Acquisition Target Set Generator. The Acquisition Target Set generator takes each specified target, along with the pumps and valves required to move material to this target and finds all dominant paths which an adversary must use to set up material flow between the elements of the target, (i.e., to set valves, turn on pumps). These adversary movement paths, which are in terms of monitor area coverage, are added to the process monitor coverage and mode information to generate a set of Acquisition Target Sets (ATS).

Figure 2.4-6 shows an example of this procedure. The access point is in area A and a valve required to move material to this point is in are B.
In plant operation mode 1, the valve is on, and in mode 2, the valve is off. The valve is monitored and the source of material, a storage tank, is monitored by a liquid level monitor.

Mode 1 has only three monitors in the Acquisition Target Set, whereas mode 2 has four monitors in use. Mode 1 has one less monitor, because it has one less area that the adversary needs to enter. We do not assume that the valve switch monitor is turned off when the valve is open. This information regarding the status of monitors is included in the Level 1.5 analysis.

The Acquisition Target Set generator also cross references ATS's, EXTS's, and ENTS's.

2.4.1.4 Output Summarizer. The Output Summarizer takes the complete sets of correlated ATS, EXTS, and ENTS as shown in Section 2.4.2 and reduces the data to the type of summary output shown in that section.


LL
TANK
M1
Access Point
LS
Valve
Area A
Area B
M2

LS = limit switch
LL = Level Monitor
M1 = Area Monitor
M2 = Area Monitor

(a) Physical Layout and Monitor Coverage

Mode 1:	M1	LL	LS	
Mode 2:	M1	LL	LS	M2

(b) Monitors in ATS

FIGURE 2.4-6: Acquisition Target Set Generation

BEST DOCUMENT AVAILABLE

2.4.2 Sample Output

An example of summary output for Level 1 is found on pages 99-101 of the microfiche. An example of the complete output is found on pages 102-106 of the microfiche in the pocket at the end of this document.

2.4.3 Code Listing

Code listing 2.4.3 is found on pages 107-236 of the microfiche in the pocket at the end of this document.

2.5 LEVEL 1.5 - WEAK COLLUSION TEST

The simple collusion analysis incorporated into the Structured Assessment Approach is called Level 1.5[8], and follows the determination of the Monitor Target Sets in Level 1. Level 1.5 performs an analysis similar to the analysis performed by the Science Applications, Inc.-developed Matrix Analysis of the Insider Threat (MAIT)[9].

2.5.1 Technical Description of Level 1.5

The Monitor Target Sets (MTS) of Level 1 describe the Material Control System coverage of all possible adversary diversion paths. A straightforward substitution of the symbols for individuals for the symbols for monitors, where those individuals either have access to or control of each of those monitors of a given MTS, yields a set of individuals who can disable that MTS. The sequence of Figures 2.5-1, 2.5-2, and 2.5-3 shows an example of the procedure for a simple case. Notice that, as before, the complete diversion path is composed of an Acquisition Target Set, an Exit Target Set, and an Entry Target Set. The simplified example shown here ignores several complications.

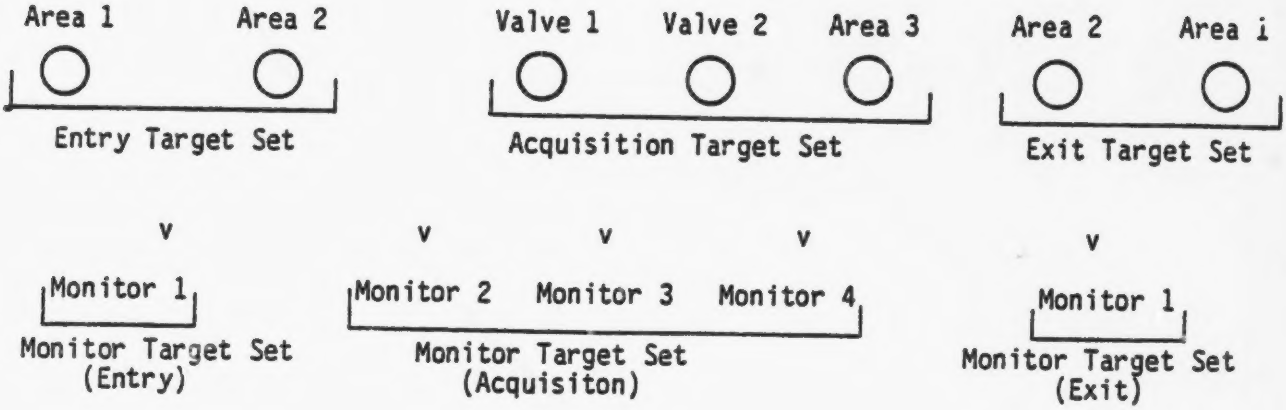

Area 1
Area 2
Valve 1
Valve 2
Area 3
Area 2
Area i
Entry Target Set
Acquisition Target Set
Exit Target Set
v
v
v
v
v
Monitor 1
Monitor 2
Monitor 3
Monitor 4
Monitor 1
Monitor Target Set (Entry)
Monitor Target Set (Acquisiton)
Monitor Target Set (Exit)

FIGURE 2.5-1: Example of Monitor Target Sets

Monitor	Individual
M1	Guard 1
M2	Guard 2
M3	Guard 2
M4	Guard 2

FIGURE 2.5-2: Access and Control List

MTS = MTS (Entry) . MTS (Acquisition) . MTS (Exit)

MTS = M1 . M2 . M3 . M4 . M1

Replace Monitor by Access or Control Individual

Compromise MTS = GUARD 1 . GUARD 2 . GUARD 2 . GUARD 1

Compromise MTS = GUARD 1 . GUARD 2

FIGURE 2.5-3: Determination of Individuals Who Can Compromise MTS

In some cases, more than one individual will have access to or control of a particular monitor. In other cases, two or more people may be required to disable or circumvent a monitor. The first case of multiple individuals with access control can be represented by the Boolean Equation

$$M1 = A + B + C + - - -$$

where

M1 is the monitor and

A, B, C, ... are the individuals who can individually defeat the monitor. If the MTS had two monitors, M1 and M2, and if M1 could be controlled by A or B and if M2 could be controlled by B or C, the Level 1.5 collusion analysis would find:

$$\begin{aligned} & MTS = M1 \,.\, M2 \\ \text{Compromise} \quad & MTS = (A+B) \cdot (B+C) \\ \text{Compromise} \quad & MTS = AB + BB + AC + BC \\ \text{Compromise} \quad & MTS = B + AC \end{aligned}$$

where a dominance argument has reduced the Boolean Equation to a minimum number of terms.

The full output from the Level 1.5 analysis code for this situation would be:

The following individuals have the ability to compromise the MTS.

Number	Individuals
1	B
2	A, C

The output means that there are two combinations of individuals which could, if in collusion, uncover a complete diversion path uncovered.

The second case described above refers to the situation where two or more individuals are required to defeat the monitor. For the present, in order to speed the implementation of Level 1.5, such a combination is designated by a new symbol which is handled as a single individual. This symbol is A/B for the case of control requiring both A and B. An example of the use of this symbol is:

Monitor M1 requires A and B to defeat it
Monitor M2 requires B or C.

The input for monitor M1 is A/B and for monitor M2 is B + C.

The output from Level 1.5 for the MTS containing M1 and M2 is then:

The following individuals have the ability to compromise the MTS:

Number	Individuals
2	A/B, B
2	A/B, C

Notice that the computer will not, at present, determine that the B of the first line is contained in the A/B symbol. This is left to the analyst to do. The processing needed to perform this step will be added later.

Another complication in the Level 1.5 analysis is the effect of plant operational mode. The set of monitors that an individual has access to and/or can control is a function of the mode of the plant. Depending on the mode, different monitors are operating, and different people control the monitors. In the locked-down mode, the portal to the Material Balance Area might not be monitored by a guard, whereas in dayshift operation it would. For example, the guard may not be able to turn off certain monitors when the facility is locked down. Thus, the Level 1.5 analysis considers the two effects of mode: the actual operational monitors, and the authorized access/control for that mode.

The collusion analysis of Level 1.5 is limited in that it considers only the direct access and/or control of monitors or procedures and does not consider indirect effects. Possibly in some situations a single individual could cause an essential support component for several monitors to fail, thus disabling those monitors. The Level 2 code expands the MTS to include these essential components and thus is required before this level of analysis. The indirect collusion analysis is called Level 3.5 and is described in a later section of this volume.

2.5.2 Sample Output

The output of Level 1.5 is of two types: a set of summary tables and a complete listing of all collusion vulnerabilities found by Level 1.5.

A sample of the summary output for Level 1.5 is on pages 237-245 of the microfiche in the pocket found at the end of this document.

A sample of the complete output of Level 1.5 is on pages 246-251 of the microfiche in the pocket at the end of this document.

2.5.3 Code Listing

The code Level 1.5 has two distinct sections: the basic Level 1.5 processor and the Level 1.5 output summarizer. These codes are listed on pages 252-289 and pages 290-298, respectively, on the microfiche in the pocket at the end of this document.

2.6 LEVEL 2/3 - ADEQUACY AND SENSITIVITY DETERMINATION

This section contains a technical description and codes of the Level 2 and Level 3 analyses. The function of Level 2 is to determine the probability of detection of adversaries on each of the Monitor Target Sets (MTS) of Level 1. Level 3 is used to determine which failures of MC&A components cause any MTS to fail.

2.6.1 Technical Description of Level 2/3

Level 2 and Level 3 are combined into a single step in the SAA[10,11]. The main activity of Level 2 is expanding the list of Monitors of the MTS of Level 1 to include the components required to support these monitors. The probability calculation of Level 2 consists of two major portions. The first is generating of the Boolean Reliability equation and the second is processing these equations into a detection probability and a MC&A System False Alarm Rate.

2.6.1.1 General Flow of Levels 2 and 3. The Level 1 analysis defines the Target Sets, along with the monitor, utility, and MC&A network coverage of these target sets. Three types of target sets have been identified: these are (1) the Acquisition Target Set, ATS, (2) the Exit Target Set, EXTS, and (3) the Entry Target Set, ENTS. Level 2 analysis uses a probability code[12] to determine the probability of detection (in most cases bounds on P_D) for the target sets assuming that all of the identified components are subject to random failure. An example of the MC&A network and associated utility network is shown in Figure 2.6-1.

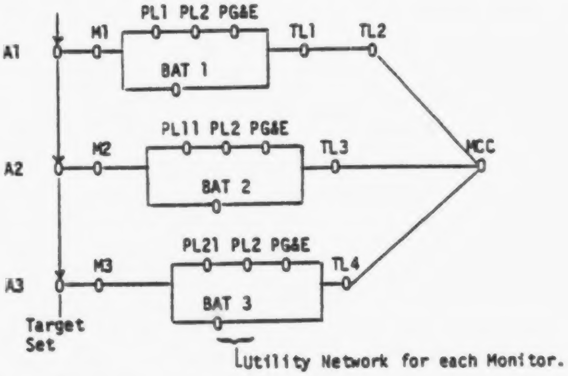


FIGURE 2.6-1: Example of Target Set Coverage Network

In order to determine the effects of a sure failure of any of the components in Figure 2.6-1, it is necessary to feed the entire Boolean equation for the coverage to the probability code. This Boolean equation is:

M1 [PL1 PL2 PG&E + BAT1] TL1 T12 MCC
+ M2 [PL11 PL2 PG&E + BAT2] TL3 MCC
+ M3 [PL21 PL2 PG&E + BAT3] TL4 MCC

The probability code is capable of doing a sensitivity analysis on this equation or network and will produce a set of detection probabilities given the failure of each component, that is

P_D (Failure of PL1)
P_D (Failure of PL2)
P_D (Failure of PG&E)
.
.
etc.

This leads to the sets of pairs:

P_D (Failure of PL1) , P (Failure of PL1)
P_D (Failure of PL2) , P (Failure of PL2)
P_D (Failure of PL3) , P (Failure of PL3)
.
.
etc.

For more detail, see Reference[13].

The problem with this type of Level 3 analysis is analyzing the full Boolean Equation for Target Set Coverage which includes (1) the coverage of the ATS, and (2) and (3) the coverage of the linked ENTS and EXTS.

This could lead to a rather complex set of paths and lengthy calculation of the probabilities. (In the example, we must consider six signal flow and utility paths with commonality between three of the paths, excluding the node marked MCC. One way, of course, to identify the critical components which should be removed is to use a cut set code such as SETS or FTAP. These codes would generate all ways to cut the information flow. For the case in the example, the minimum cut set codes would find that, excluding MCC again, there is no single component whose failure will disable the network. However, there are 3 cut sets of order 3. These are

TL1, TL3, TL4
TL2, TL3, TL4
M1, M2, M3

The complete failure of the network thus requires at least three component failures.

In order to avoid the need for a cut set or factorization code and the lengthy attendant probability calculation at this time, the Level 3 failure analysis at present is limited to a single component failure.

The flowchart for Levels 2 and 3 is given in Figure 2.6-2. The plan behind the flowchart of Figure 2.6-2 is to make repeated passes at comparing a bound on detection probability to the NRC Adequacy criteria. The calculation performed for each pass is exact, that is no approximations are made. Successive passes use more and more of the networks for the Monitor Target Sets. These bounding passes are made in order to limit the calculations required by the probability code on complex reliability networks.

Figure 2.6-2 shows the flowchart for Levels 2 and 3 as currently implemented. The notes explain the functions shown in the figure.

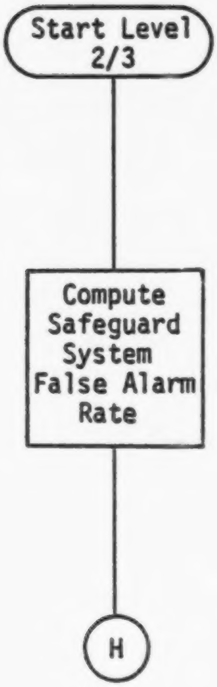

Start Level 2/3
Compute Safeguard System False Alarm Rate
H

FIGURE 2.6-2: Flowchart for Level 2 and Level 3

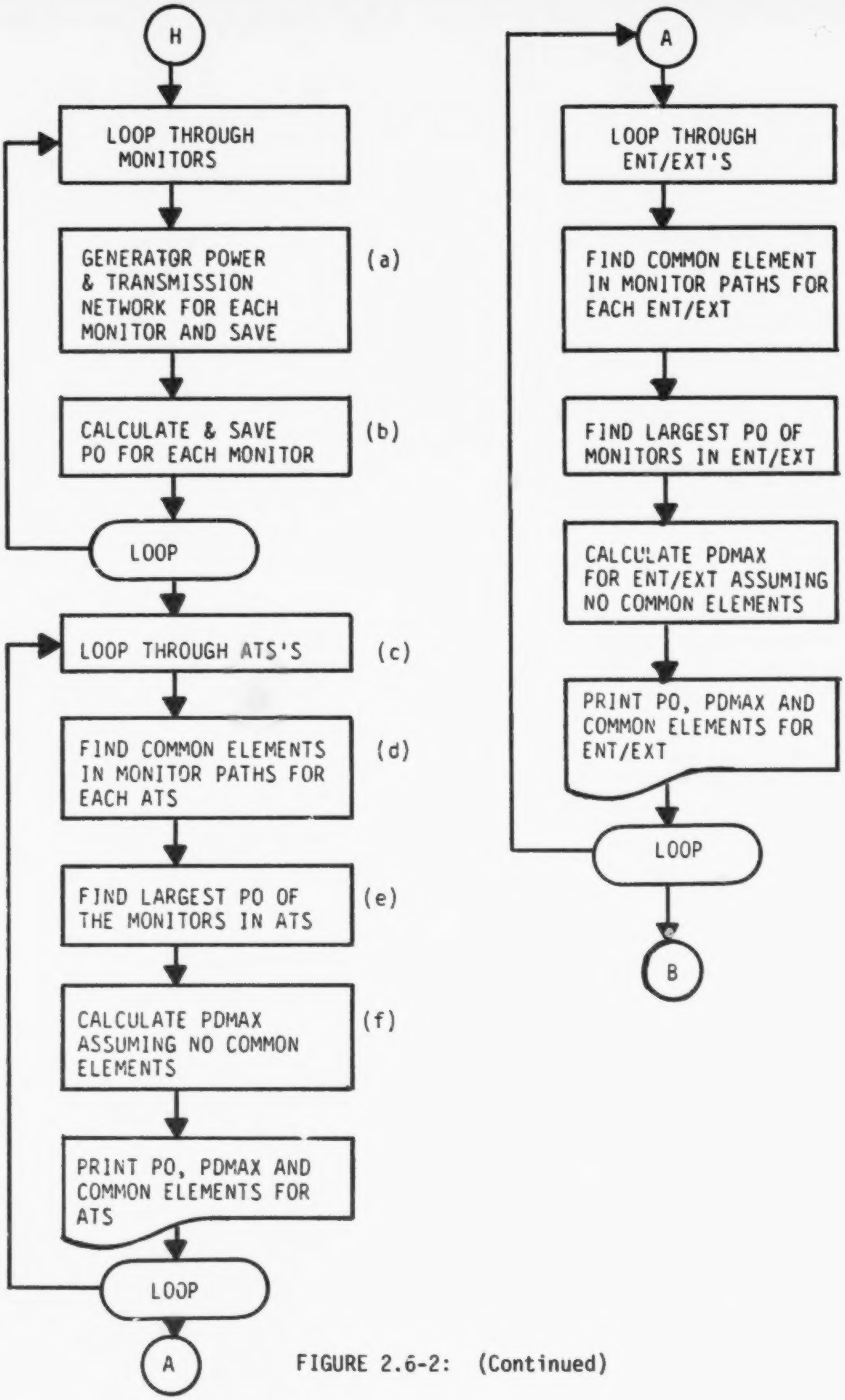

H
LOOP THROUGH MONITORS
GENERATOR POWER & TRANSMISSION NETWORK FOR EACH MONITOR AND SAVE
(a)
CALCULATE & SAVE PO FOR EACH MONITOR
(b)
LOOP
LOOP THROUGH ATS'S
(c)
FIND COMMON ELEMENTS IN MONITOR PATHS FOR EACH ATS
(d)
FIND LARGEST PO OF THE MONITORS IN ATS
(e)
CALCULATE PDMAX ASSUMING NO COMMON ELEMENTS
(f)
PRINT PO, PDMAX AND COMMON ELEMENTS FOR ATS
LOOP
A
A
LOOP THROUGH ENT/EXT'S
FIND COMMON ELEMENT IN MONITOR PATHS FOR EACH ENT/EXT
FIND LARGEST PO OF MONITORS IN ENT/EXT
CALCULATE PDMAX FOR ENT/EXT ASSUMING NO COMMON ELEMENTS
PRINT PO, PDMAX AND COMMON ELEMENTS FOR ENT/EXT
LOOP
B

FIGURE 2.6-2: (Continued)

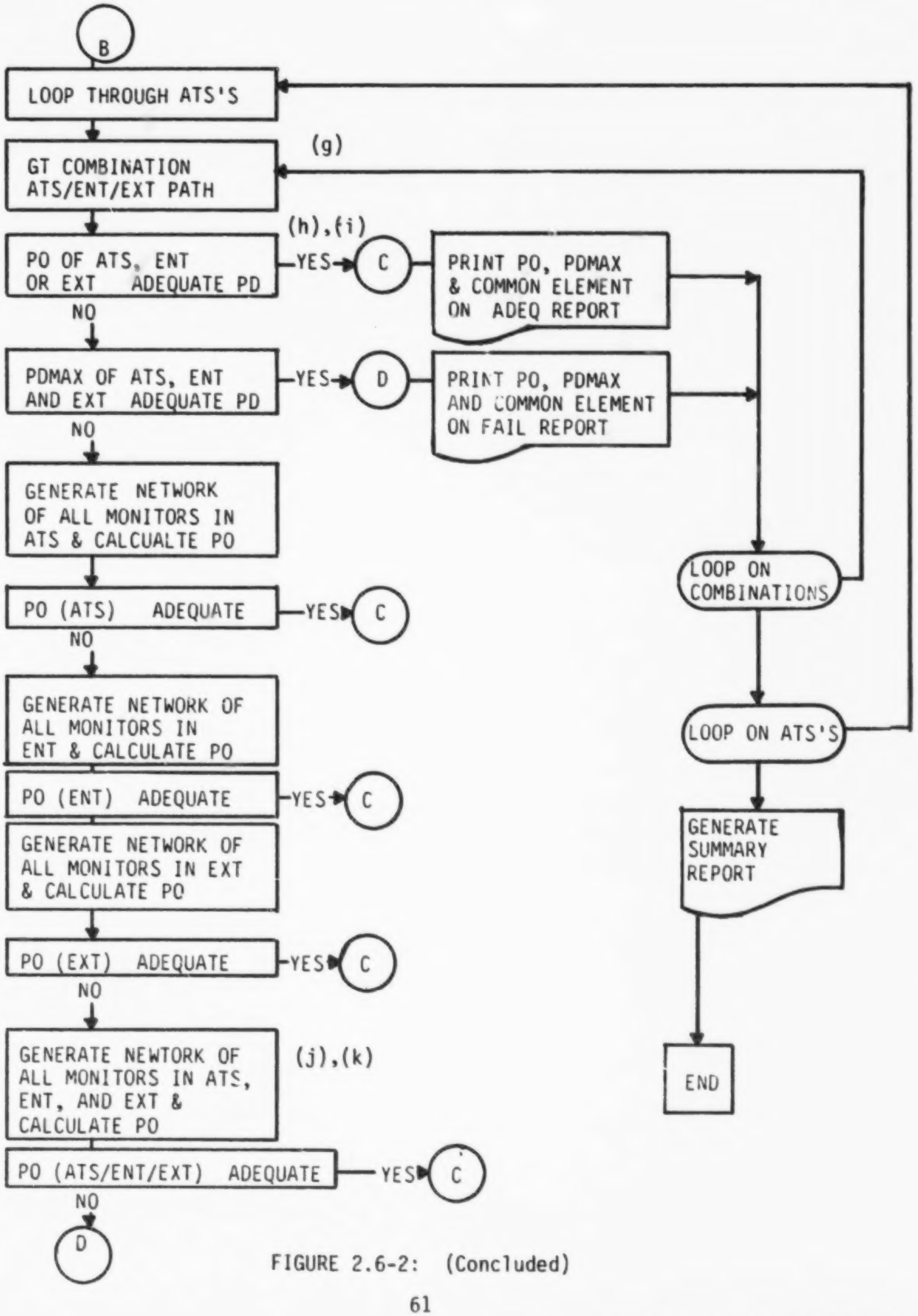

B
LOOP THROUGH ATS'S
(g)
GT COMBINATION ATS/ENT/EXT PATH
(h),(i)
PO OF ATS, ENT OR EXT ADEQUATE PD
YES
C
PRINT PO, PDMAX & COMMON ELEMENT ON ADEQ REPORT
NO
PDMAX OF ATS, ENT AND EXT ADEQUATE PD
YES
D
PRINT PO, PDMAX AND COMMON ELEMENT ON FAIL REPORT
NO
GENERATE NETWORK OF ALL MONITORS IN ATS & CALCUALTE PO
LOOP ON COMBINATIONS
PO (ATS) ADEQUATE
YES
C
NO
GENERATE NETWORK OF ALL MONITORS IN ENT & CALCULATE PO
LOOP ON ATS'S
PO (ENT) ADEQUATE
YES
C
GENERATE NETWORK OF ALL MONITORS IN EXT & CALCULATE PO
GENERATE SUMMARY REPORT
PO (EXT) ADEQUATE
YES
C
NO
GENERATE NEWTORK OF ALL MONITORS IN ATS, ENT, AND EXT & CALCULATE PO
(j),(k)
END
PO (ATS/ENT/EXT) ADEQUATE
YES
C
NO
D

FIGURE 2.6-2: (Concluded)

Notes to Figure 2.6-2

(a) Each monitor which is used will therefore have an equation attached to it.

(b) The probability is calculated in a subroutine supplied by Applied Decision Analysis, Inc., and described in Section 2.6.1.2 which computes the probability of a given path being operational. This probability is called P_σ.

(c) ATS = Acquisition Target Set, EXTS = Exit Target Set, ENTS = Entry Target Set. A list of the monitors is then created.

(d) This step forms the basis for a calculation of a lower bound for the detection probability for a given Monitor Target Set. The highest P_σ (monitor) $\leq P_\sigma$ (entire MTS). Thus, if this P_σ (monitor $\geq$ NRC Adequacy Criteria, the entire MTS is adequately covered. This bounding procedure is performed to reduce the "size" of the probability calculation.

(f) This calculation determines the maximum possible detection probability for a given MTS. The calculation is

$$P_{D_{MAX}} \text{ (MTS)} = 1 - \prod_{L=1}^{N} 1-P_D \text{ (monitor i) for a given MTS.}$$

N is the number of monitors in the set and P_D (monitor i) is the probability that monitor i will detect its stimulus. This detection probability includes the reliability of the transmission and power path for the monitor. This, of course, assumes each monitor is independent of all others in the MTS. Hence if $P_{D_{MAX}} <$ (MTS) NRC Adequacy Criteria, the MTS is not adequately covered.

(g) This printout requires the use of the ATS, EXTS, ENTS link lists. The P_o and P_{MAX} printed in this table are

$$P_o(A/E/E) = MAX \left\{ P_o \text{ (ATS)}, P_o \text{ (EXTS)}, P_o \text{ (ENTS)} \right\}$$

$$P_{MAX}(A/E/E) = 1 - (1-P_{MAX}(ATS)) \cdot (1-P_{MAX}(EXTS)) \cdot (1-P_{MAX}(ENTS))$$

This step determines if the total Diversion Path (composed of ATS, EXTS, and ENTS) is adequately covered. The test is

$$\text{if } P_o \ (A/E/E) > NRC \longrightarrow \text{adequate}$$

$$\text{if } P_{MAX} \ (A/E/E) < NRC \longrightarrow \text{not adequate}$$

See attached sample output for Levels II and III.

(h) There will, in general, be several EXTS, ENTS sets with each ATS.

(i) This determines that at least one phase of diversion scenario is covered.

(j) This step generates the full MTS for each MAYBE Acquisition Target Set, including all monitors and their power and transmission paths.

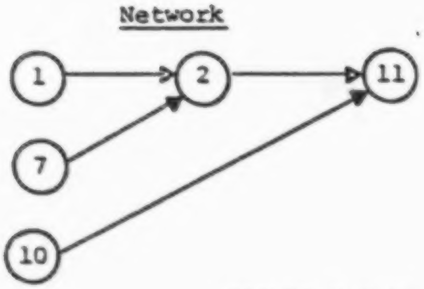
(k) The timer is used to prevent the PROB code from using an excessive amount of time. If excessive time is used, the following message will be printed:

ATS N Requires Excessive Probability Calculation Time

2.6.1.2 Probability Calculation. The probability of the detection calculation specified in the flow chart (Figure 2.6-2) of the previous section is performed by a code called PROB[12]. This code is used to convert the Boolean Expression which represents either part of or all of the complete Monitor Target Sets of Level 2 into a detection probability, including an availability for that MTS. This code is described below along with a special code which was designed to minimize the computational time requirements[14]. In addition, the technique to determine the system false alarm rate is described.

2.6.1.2.1 Probability Code. The purpose of the probability computer algorithm is to compute the exact calculation of the probability of detection for a path representation of a given Boolean expression and a given set of component event probability data. Both the data and the expression represent the specific conditions associated with the adversary type, the monitor target set, and facility operating mode.

The simple network shown in Figure 2.6-3 is used to illustrate the solution approach. In this example, the path representation of the detection event D for the following simple network is the logical sum of the three paths:



Path Representation

$D = P1 + P2 + P3$

$P1 = (1)\cdot(2)\cdot(11)$
$P2 = (7)\cdot(2)\cdot(11)$
$P3 = (10)\cdot(11)$

FIGURE 2.6-3: Example Network

The Level 2 code defines a detection event in terms of a collection of paths. Each path is described by a list of components. A collection of paths that define a detection event is input into the PROB algorithm for probabilistic processing.

The probability of detection for this example is calculated by the following algebraic formula:

$$p(D) = p(P1) + p(P2) + p(P3) - p(P1\ P2) - p(P1\ P3) - p(P1\ P3) + p(P1\ P2\ P3)$$

The path probabilities are calculated from the product of the component event probabilities input to the code. For example:

$$p(P1) = p(1)\ p(2)\ p(11)$$

The code extends this algebraic formula to accommodate as many paths as will fit in the computing system's memory capacity or the analyst has time to wait for. The bounding test for PROB is discussed in Section 2.6.1.3.

This formula is implemented by the procedure outlined in Figure 2.6-4. Each component has an integer identifying number and a probability. Each path is assigned a bit vector in the array PATH. Path components are represented by a 1 in a specific bit location within the vector. A 0 for a component means that it is not contained in the path.

Graphically, the example above would be represented as shown in Figure 2.6-5.

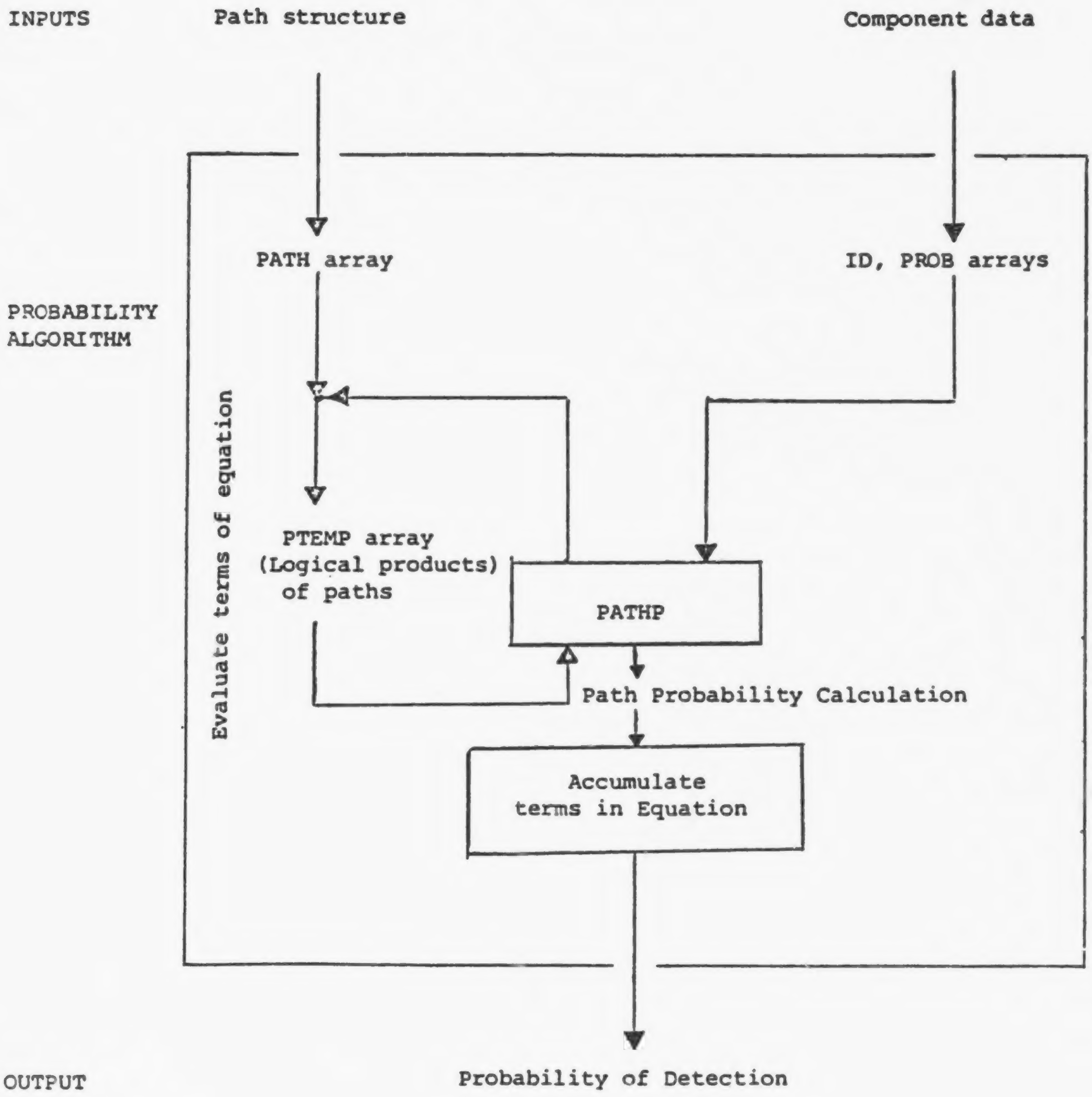

INPUTS
Path structure
Component data
PATH array
ID, PROB arrays
PROBABILITY ALGORITHM
Evaluate terms of equation
PTEMP array
(Logical products)
of paths
PATHP
Path Probability Calculation
Accumulate
terms in Equation
OUTPUT
Probability of Detection

FIGURE 2.6-4: Conceptual Structure of Probability Algorithm

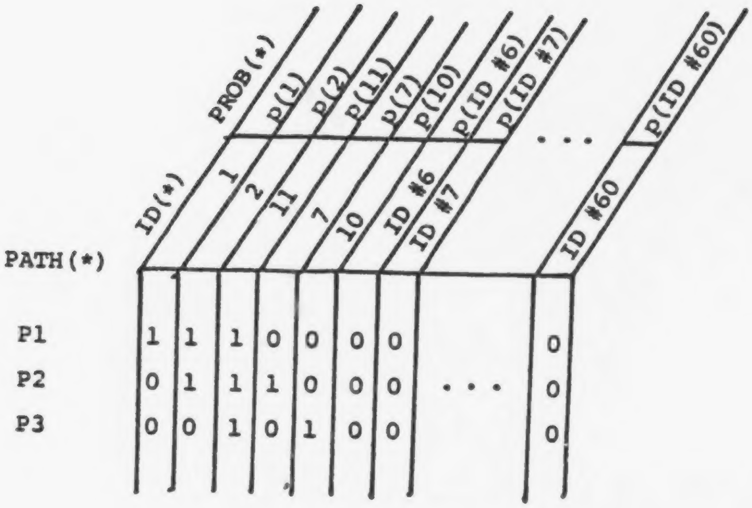


FIGURE 2.6-5: Bit Vector Representation of MTS

Each path probability is then calculated in Function PATHP as the product of the probabilities of the components corresponding to the 1 bits in the path word. The sum of the path probabilities is stored. The code then continues to find the remaining probabilities in the equation for the probability of detection.

Probabilities for each member of the set of logical products (e.g., P1 P2) with two paths are evaluated. The logical products of two paths are also stored as bit vectors with 1 or 0 representing whether or not a component is included in result in the array PTEMP. The probability of each logical product is then calculated by the function PATHP and subtracted from the sum of path probabilities. The logical product not containing path P1 is saved (P2 P3) for generating the next set of logical products. In this example, only one member (P1 P2 P3) is in this set. The probability of this logical product is then added to the others to yield the exact calculation of the probability of detection.

A general calculation with N paths repeats these procedures for N-1 sets of logical products. The K^{th} set of logical products contains all combinations of paths that can occur taken K paths at a time, excluding the first path. The size of K^{th} set is determined by the number of ways K paths can be combined out of a total of N-1 paths:

$$\begin{array}{l}\text{Number of logical}\\ \text{products stored in}\\ K^{th}\text{ set}\end{array} = \frac{(N-1)\,!}{(K!\,(N-K-1)!)}$$

To generate the K+1 set of logical products, path(1) through path(N-K) are each used with the K^{th} set of logical products. Path(1) forms the logical product with all members of the set. Path(2) will form a logical product with all but the first (C_K^{N-1} - C_K^{N-2} paths. The path(L), where L is not larger than N-K, will form logical products with the same elements in the K^{th} set as path (L-1), except for the first (C_K^{N-L+1} - C_K^{N-L}) elements. This rule forms a recursive relation that is easily programmed to evaluate the probability of detection for any number of paths within limits of the code.

The flowchart for the calculations with the probability subroutine is shown in Figures 2.6-6, 2.6-7, and 2.6-8.

2.6.1.2.2 Bounding Tests. We encountered a problem attempting to use the algorithm which computes the probability of detection for the complete set of signal paths including utilities in Level 2 of the SAA. The code began to use excessive amounts of computer time and storage when 12 event paths were passed to it. In order to temporarily circumvent this problem, a step has been added before the event paths are transferred to the probability code. This section describes this temporary fix.

Rather than sending the Probability Code the entire Boolean equation which describes a monitor, or a monitor target set, along with its power and transmission paths, several bounding tests are made.

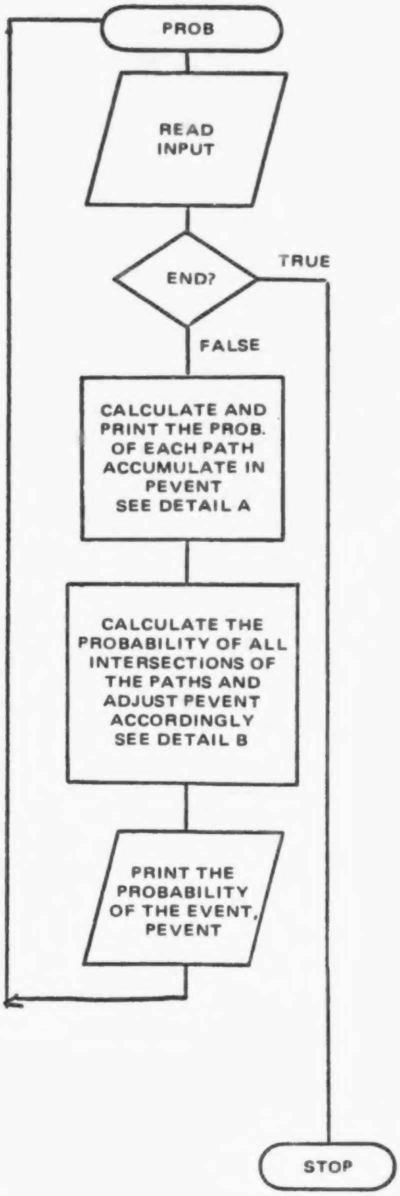

PROB
READ INPUT
END?
TRUE
FALSE
CALCULATE AND PRINT THE PROB. OF EACH PATH ACCUMULATE IN PEVENT SEE DETAIL A
CALCULATE THE PROBABILITY OF ALL INTERSECTIONS OF THE PATHS AND ADJUST PEVENT ACCORDINGLY SEE DETAIL B
PRINT THE PROBABILITY OF THE EVENT, PEVENT
STOP

FIGURE 2.6-6: Probability Calculations For PROB

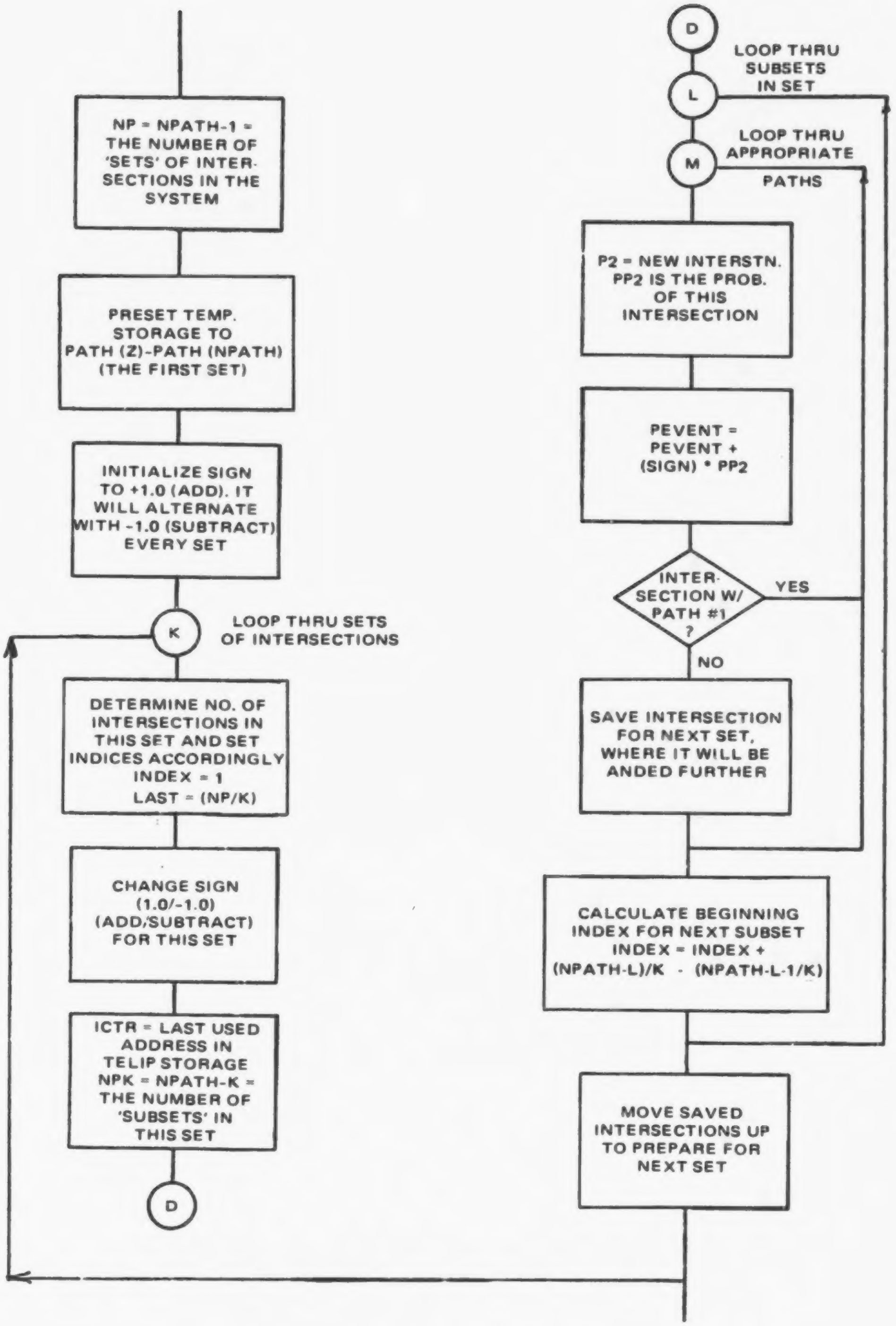

NP = NPATH-1 = THE NUMBER OF 'SETS' OF INTERSECTIONS IN THE SYSTEM
PRESET TEMP. STORAGE TO PATH (Z)-PATH (NPATH) (THE FIRST SET)
INITIALIZE SIGN TO +1.0 (ADD). IT WILL ALTERNATE WITH -1.0 (SUBTRACT) EVERY SET
K
LOOP THRU SETS OF INTERSECTIONS
DETERMINE NO. OF INTERSECTIONS IN THIS SET AND SET INDICES ACCORDINGLY INDEX = 1 LAST = (NP/K)
CHANGE SIGN (1.0/-1.0) (ADD/SUBTRACT) FOR THIS SET
ICTR = LAST USED ADDRESS IN TELIP STORAGE NPK = NPATH-K = THE NUMBER OF 'SUBSETS' IN THIS SET
D
D
L
LOOP THRU SUBSETS IN SET
M
LOOP THRU APPROPRIATE PATHS
P2 = NEW INTERSTN. PP2 IS THE PROB. OF THIS INTERSECTION
PEVENT = PEVENT + (SIGN) * PP2
INTERSECTION W/ PATH #1 ?
YES
NO
SAVE INTERSECTION FOR NEXT SET, WHERE IT WILL BE ANDED FURTHER
CALCULATE BEGINNING INDEX FOR NEXT SUBSET INDEX = INDEX + (NPATH-L)/K - (NPATH-L-1/K)
MOVE SAVED INTERSECTIONS UP TO PREPARE FOR NEXT SET

FIGURE 2.6-7: Detail B For Prob

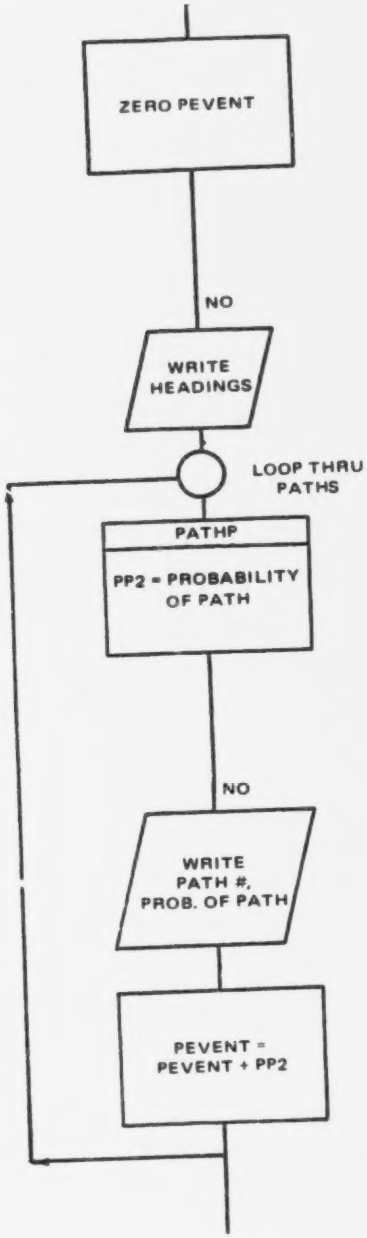

ZERO PEVENT
NO
WRITE HEADINGS
LOOP THRU PATHS
PATHP
PP2 = PROBABILITY OF PATH
NO
WRITE PATH #, PROB. OF PATH
PEVENT = PEVENT + PP2

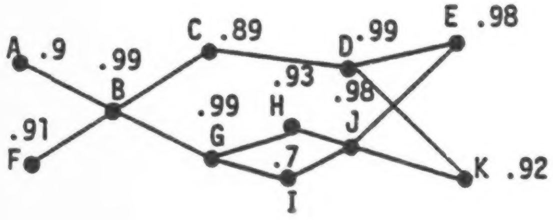
FIGURE 2.6-8: Detail A for PROB

First, the entire set of Boolean event equations which describes the monitor or monitor target set is checked for a common factor. If a common factor is found, the probability of this common factor is compared to the NRC Adequacy Criterion. If the value of the probability of this factor is less than the value specified in the criterion, then the probability calculation is terminated, since the probability for the entire equation must be lower. If the equation passes this test, then the probability of each term in the equation is calculated separately. If the best of these terms is good enough to pass NRC criteria, then the entire path is good enough. If not, the n best terms are calculated together by the probability code. These n terms are chosen in the following manner:

1) The best 2N terms are sorted in order, term 1 having the highest probability.

2) Terms 2 through N are compared to each term N+1 through 2N. If the difference between their probabilities is less than .01, they are each compared against term #1, and the ones with the least common elements with term 1 are chosen.

Consider the following example:



N = 4

The graph has the following terms and probabilities

Terms	Pd	Choice	Terms	Pd	Choice
ABCDE	.730	⑥	ABGIJE	.568	
ABCDK	.722	⑧	ABGIJK	.562	
FBCDE	.738	⑤	FBGHJK	.747	②
FBCDK	.730	⑦	FBGHJE	.747	③
ABGHJE	.747	①	FBGIJK	.563	
ABGHJK	.739	④	FBGIJE	.569	

Comparing probabilities of term 2, 3, 4 against 5, 6, 7, and 8 leaves 2 and 3 in place since they are both .74. Comparing 4 and 5, that is, comparing the terms ABGHJK and ABGHJE versus FBCDE and ABGHJE

④	.739	ABGHJK	①	ABGHJE
⑤	.738	FBCDE		

⑤ replaces ④ since it has fewer factors in common with ① .

The new ④ compares against ⑥

④	.738	FBCDE	①	ABGHJE
⑥	.730	ABCDE		

④ still has fewer factors in common with ① .

④ compares against ⑦

④	.738	FBCDE	①	ABGHJE
⑦	.730	FBCDK		

⑦ replaces ④ since it has fewer factors in common with ① .

Term ⑧ is not considered since it is .72. The four terms sent to the probability code are:

① ABGHJE
② FBGHJK
③ FBGHJE
④ FBCDK

2.6.1.2.3 Safeguard System False Alarm Rate. One of the statistics gathered in the LSD is the False Alarm Rate (FAR) or mean time between False Alarms (MTBFA) for each monitor[15]. The data are gathered with the intention of calculating an overall false alarm rate for the entire safeguard detection system.

The contribution that each monitor i makes to the overall false alarm rate is:

$$FAR_i = \frac{X_i}{Y_i} \quad \frac{1}{MTBFA_i}$$

where

X_i is the probability number currently calculated for each monitor i in the algorithm.

Y_i is the "local" probability of detection (PD) for monitor i, excluding the availability number for monitor i.

$MTBFA_i$ is the mean time between false alarms for monitor i (in hours).

In other words, the FAR_i is the probability that monitor i has an available path to the response nodes times the local false alarm rate $\frac{1}{MTBFA}$ for the monitor. The total False Alarm Rate for the system is then:

$$SFAR = \sum_{i=1}^{I} FAR_i, \qquad I = \text{total number of monitors}$$

The expected number of false alarms each hour is seen from the physical security alarm stations or response nodes. This number (SFAR) would then be a direct output.

The above equation for SFAR is valid only if false alarms from different monitors are distinguishable. That is, each alarm is unique for each monitor in the security stations, and each alarm is responded to uniquely. This is taken to be a fair and conservative assumption at this time. Any overlapping in time of false alarms or other potential correlating factors would lower the safeguard system's overall false alarm rate.

2.6.2 Sample Output

Typical output from Level 2 is shown on the microfiche. The Level 2/3 Output Results for Adequately Covered Diversion paths is on pages 308-312 of the microfiche and contains information on all complete diversion paths which pass the NRC adequacy criteria. Listed are the overall system false alarm rate, and individual detection probabilities for each phase of the diversion scenario (ATS, EXTS/ENTS) for each operational mode of the facility. The combined P_D for the entire diversion scenario is also listed.

Pages 313-317 of the microfiche contain a sample of the Level 2/3 output for Monitor Target Sets which fail the NRC adequacy criteria.

Summary output from Level 3 is found on pages 299-307 of the microfiche.

2.6.3 Code Listing

Level 2/3 Assurance and Sensitivity Analysis Code

The code listing 2.6.3 is found on pages 318-397 of the microfiche in the pocket at the end of this document.

2.7 LEVEL 3.5 - SECOND PHASE COLLUSION TEST

The collusion analysis of Level 1.5 is equivalent to the MAIT analysis developed by SAI. Unfortunately, both analyses suffer from a rather restricted point of view. They look only at the effect of an individual who has authorized access to monitors within a monitor target set, MTS. They do not investigate if some other individual or individuals could disable the coverage on an MTS by disabling some system component other than the monitor* itself. For example, if an adversary could turn off the component by interrupting the public utility power, Level 1.5 would not credit the adversary with the ability to uncover MTS's which depended on these lights. Level 3.5 is designed to address this shortcoming by duplicating the analysis of Level 1.5 on the complete monitor paths for each MTS of Level 1[16]. The complete monitor paths include all elements which support the operation of each monitor in each MTS.

2.7.1 Technical Description of Level 3.5

The Level 2/3 code determines the complete Boolean Reliability Equation for each MTS. For the example shown in Figure 2.7-1, this is:

$$\begin{aligned} MTS1 = {} & M1 \bullet [BAT1 \bullet PL3 \bullet PL2 \bullet PG\&E] \bullet TL1 \bullet TL2 \bullet R1 \\ & + M2 \bullet [BAT2 \bullet PL1 \bullet PL2 \bullet PG\&E] \bullet TL3 \bullet TL4 \bullet R1 \end{aligned}$$

---

*Monitor in this context could also be a procedure such as a search at the portal to a Material Balance Area. In this case, the Level 1.5 analysis would have indicated that the portal guard would be a part of a collusion set which uncovered a MTS.

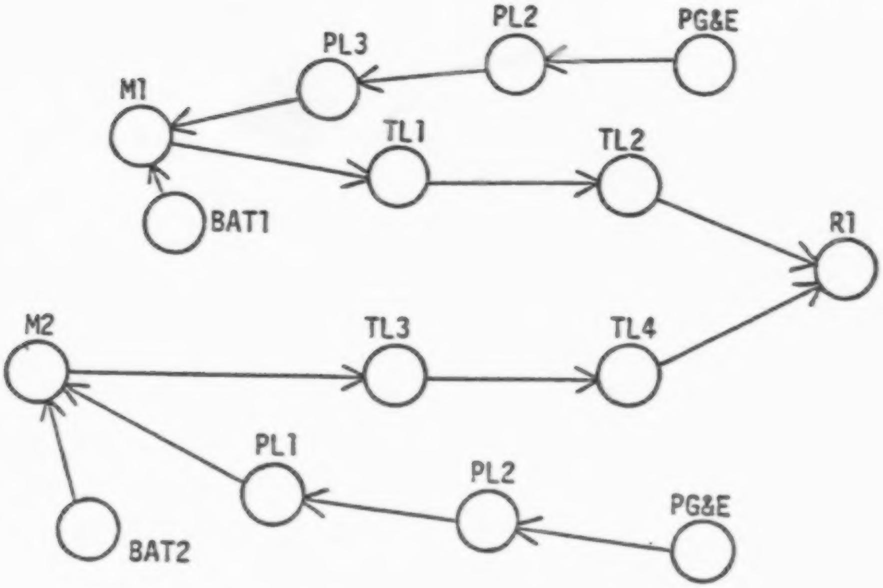


FIGURE 2.7-1: Monitor Target Set With Support Systems

By using the same type of authorized access lists as those used in Level 1.5, we can determine which combinations of individuals can disable this MTS. As in Level 1.5, a list of individuals with their authorized accesses is required. Table 2.7-1 is this list for the hypothetical facility which has MTS1.

	MM1	MM2	GPAS	GMBA2
M1	1			1
M2		1		
BAT1	1			
BAT2		1		
PL1	1			
PL2	1			
PL3		1		
PG&E		1		
TL1	1			
TL2		1		
TL3	1			
TL4		1		
R1			1	

Notes
MM1 - Maintenance Man 1
MM2 - Maintenance Man 2
GPAS - Guard-Primary Alarm Station

GMBA2 - Guard - MBA-2

TABLE 2.7-1: Authorized Access List

Substitution of these authorized accesses into the MTS equation for failure yields the Level 3.5 collusion results.

The MTS failure equation is generated by using DeMorgan's Law ($\overline{A + B} = \overline{A} \bullet \overline{B}$) on the MTS equation. For the example, the failure equation is

$$\begin{aligned} \mathrm{FMTS1} = & \left[\overline{\mathrm{M1}} + \overline{\mathrm{BAT1}} \quad (\overline{\mathrm{PL3}} \bullet \overline{\mathrm{PL2}} + \overline{\mathrm{PG\&E}}) \quad + \overline{\mathrm{TL1}} + \overline{\mathrm{TL2}} + \overline{\mathrm{R1}}\right] \\ \bullet & \left[\overline{\mathrm{M2}} + \overline{\mathrm{BAT2}} \quad (\overline{\mathrm{PL1}} \bullet \overline{\mathrm{PL2}} + \overline{\mathrm{PG\&E}}) \quad + \overline{\mathrm{TL3}} = \overline{\mathrm{TL4}} + \overline{\mathrm{R1}}\right] \end{aligned}$$

The following result is obtained for the example by substituting in the authorized access list.

$$\begin{aligned} \mathrm{FMTS1} = & \left.\begin{aligned} & \big[(\mathrm{MM1} + \mathrm{GMBA2}) + \mathrm{MM1} \bullet (\mathrm{MM2} + \mathrm{MM1} + \mathrm{MM2}) \\ & + (\mathrm{MM1}) + (\mathrm{MM1}) + (\mathrm{GPAS})\big] \bullet \end{aligned}\right\} \mathrm{M1} \\ & \left.\begin{aligned} & \big[(\mathrm{MM2}) + \mathrm{MM2} \bullet (\mathrm{MM2} + \mathrm{MM1} + \mathrm{MM2}) \\ & + (\mathrm{MM1}) + (\mathrm{MM2}) + (\mathrm{GPAS})\big] \end{aligned}\right\} \mathrm{M2} \end{aligned}$$

This Boolean equation specified how to disable the Monitor Target Set. The above equation can be expanded out to yield the collusion sets.

$$\begin{aligned} \mathrm{FMTS1} &= [\mathrm{MM1} + \mathrm{GMBA2} + (\mathrm{MM1} \bullet \mathrm{MM2} + \mathrm{MM1} + \mathrm{MM1} \bullet \mathrm{MM2}) + \mathrm{MM1} + \mathrm{MM1} + \mathrm{GPAS}] \bullet \\ &\quad [\mathrm{MM2} + (\mathrm{MM2} + \mathrm{MM1} \bullet \mathrm{MM2} + \mathrm{MM2}) + \mathrm{MM1} + \mathrm{MM2} + \mathrm{GPAS}] \\ \mathrm{FMTS1} &= [\mathrm{MM1} + \mathrm{GMBA2} + \mathrm{MM1} \bullet \mathrm{MM2} + \mathrm{GPAS}] \bullet \\ &\quad [\mathrm{MM2} + \mathrm{MM1} \bullet \mathrm{MM2} + \mathrm{MM1} + \mathrm{GPAS}] \\ \mathrm{FMTS1} &= [\mathrm{MM1} + \mathrm{GMBA2} + \mathrm{GPAS}] \bullet \\ &\quad [\mathrm{MM2} + \mathrm{MM1} + \mathrm{GPAS}] \qquad \text{Using dominance argument} \\ \mathrm{FMTS1} &= \mathrm{MM1} \bullet \mathrm{MM2} + \mathrm{MM1} + \mathrm{MM1} \bullet \mathrm{GPAS} \\ &\quad + \mathrm{GMBA2} \bullet \mathrm{MM2} + \mathrm{GMBA2} \bullet \mathrm{MM1} + \mathrm{GMBA2} \bullet \mathrm{GPAS} \\ &\quad + \mathrm{GPAS} \bullet \mathrm{MM2} + \mathrm{GPAS} \bullet \mathrm{MM1} + \mathrm{GPAS} \\ \mathrm{FMTS1} &= \mathrm{MM1} + \mathrm{GPAS} + \mathrm{MM2} \bullet \mathrm{GMBA2} \end{aligned}$$

This is shown in Table 2.7-3 in the format used for the Level 1.5 output.

```
MTS  ACTIVE  MONITORS  THIS  MODE
 1     M1       M2

NUMBER AND COMBINATION OF PERSONNEL WHO HAVE THE ABILITY TO
COMPROMISE THIS MONITOR SET
  1       MM1
  1       GPAS
  2       MM2      GMBA2
```

Table 2.7-2: "Computer Output" for Level 3.5

2.7.2 Sample Output

Level 3.5 is not currently implemented in the SAA. Implementation is scheduled to begin in October 1979.

2.7.3 Code Listing

Level 3.5 is not currently implemented in the SAA. Implementation is scheduled to begin in October 1979.

2.8 LEVEL 4 - FULL TAMPERING VULNERABILITY

The purpose of Level 4 is to determine the vulnerability of both the material control and material accounting systems to sophisticated adversary tampering. This tampering includes:

(1) modifying the state of Material Control and Accounting (MC&A) Systems' monitors by either authorized or indirect control

(2) falsifying of information within the Material Accounting System to delay or inhibit detecting material loss during physical inventories

(3) circumventing of Safeguard System procedures which allow the undetected removal of material.

In general, Level 4 allows the complete in-depth determination of all possible ways to defeat the Safeguard System including use of stealth and deceit.

In Level 4 analysis is based on the solution of a set of Boolean equations which describe the Material Control and Material Accounting Systems. The solution technique is based on graph theory and, in particular, reachability calculation, and is described in detail in References 17 and 18. This section of Volume 4 presents a description of the general techniques used in Level 4, including modeling considerations, specification to Material Control and Material Accounting Systems, a description of the Boolean Equation Processor (CLAMOR), sample output from a Level 4 analysis of a Material Accounting System, and finally a source code listing of CLAMOR.

2.8.1 Technical Description of Level 4

2.8.1.1 General Technique Description. A network which describes the MC&A system and physical plant network[17] is constructed in the manner described in Volume 1 of this report and Reference 18. The input nodes to this network are:

1. Plant Personnel
2. Component Locations

The general system element takes the form of a Modified Petri transition (or Logical AND) and is shown in Figure 2.8-1[19].

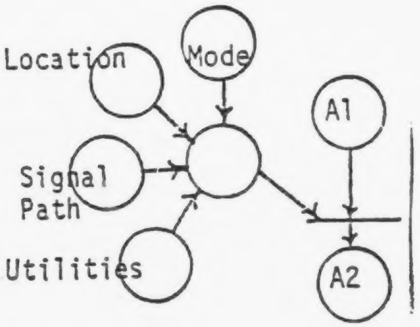


FIGURE 2.8-1: General System Element

Most of the nodes will be connected to other nodes in the MC&A system and the physical plant[19]. The Mode node is a switch which disables the monitor for specific operational modes. Thus, these nodes are also inputs to the network. The unit models are interconnected via the area, MC&A, utility, and process flow adjacency matrices. Connecting these unit models leads to a large logical network, or Modified Petri Net with inputs and outputs. This is shown schematically in Figure 2.8-2:

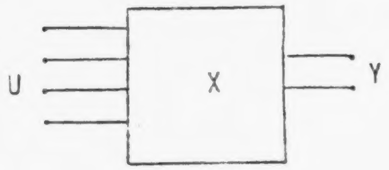


FIGURE 2.8-2: General Functional Network

where X represents the network, U the input, and Y the outputs.

The problem in Level 4 is to solve the set of Boolean equations:

$$Y = F(U, X)$$

which represents the Material Control and Material Accounting System. As stated above, this will determine which input conditions lead to a specific output. In the case of a system where personnel can defeat the MC or MA system via tampering, the solution to the equation will take the form

$$Y = P_1 P_2 + P_2 P_3$$

where P_i are personnel in the system adversary. In general the solution to defeat of the MC or MA system will be of the form: $Y = f(L_i, P_i)$; P_i are personnel and L_i are locations of components.

2.8.1.2 Material Control System Analysis. In order to illustrate the application of the technique, two examples will be presented, one for Material Control and one for Material Accounting.

The first example is shown in Figure 2.8-3.

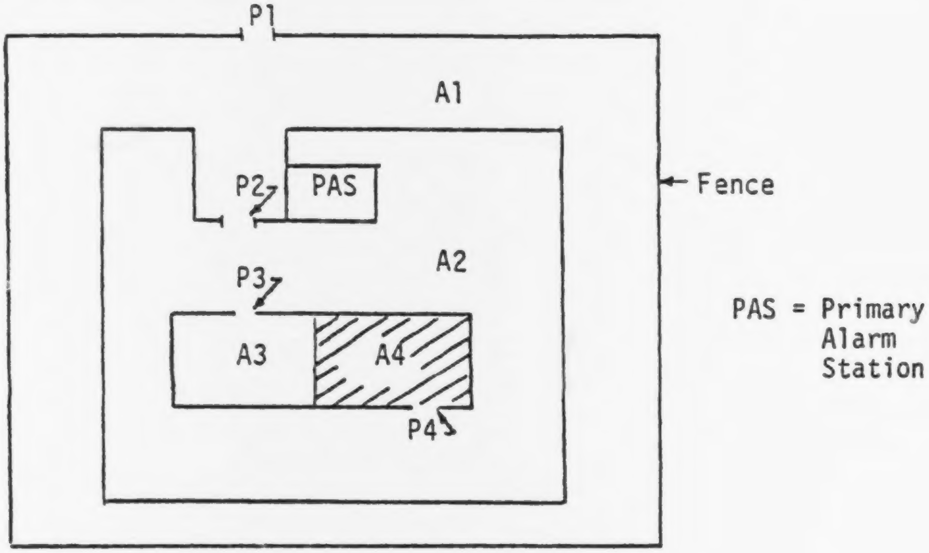


FIGURE 2.8-3: Simplicity Uranium Corporation

We assume Special Nuclear Material (SNM) is present in area A4 and the adversary will start either outside the outer perimeter or in area A4. The goal of the adversary in the first case is to penetrate from the outside (AO) to the storage area (A4). In the second case, the goal is to move from area A4 to AO. In the terminology of this paper, the two cases can be described as computing:

(1) Reachability from Ao to A4, and
(2) Reachability from A4 to Ao.

The area adjacency matrix for the example is given in Figure 2.8.4 with an area graph shown in Figure 2.8.5. In the adjacency matrix and the graph, a symbol B indicates that the areas are adjacent but separated by a barrier, which may or may not be monitored.

	A_0	P_1	A_1	P_2	PAS	P3	A2	A3	A4	P4
A_0	0	1	B	0	0	0	0	0	0	0
P_1	1	0	1	0	0	0	0	0	0	0
A_1	B	1	0	1	B	0	B	0	0	0
P2	0	0	1	0	0	0	1	0	0	0
PAS	0	0	B	0	0	0	B	0	0	0
P3	0	0	0	0	0	0	1	1	0	0
A2	0	0	B	1	B	1	0	B	B	1
A3	0	0	0	0	0	1	B	0	1	0
A4	0	0	0	0	0	0	B	1	0	0
P4	0	0	0	0	0	0	1	0	1	0

FIGURE 2.8-4: Area Adjacency Matrix for Example

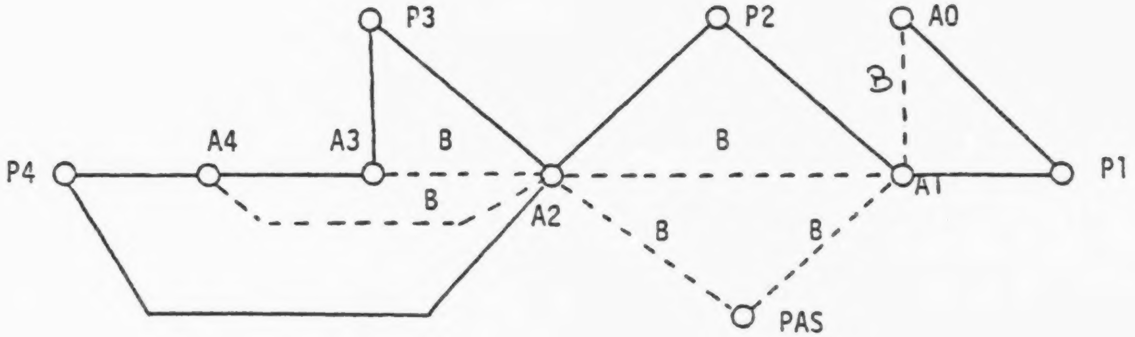

P3
P2
A0
B
A4
A3
B
B
P4
A1
P1
B
A2
B
B
PAS

FIGURE 2.8-5: Area Graph for Example

Dashed lines indicate physical adjacencies which are blocked by Barriers.

The Field-of-View Matrix or Coverage Matrix which describes which monitors cover which matrix is shown in Figure 2.8-6

	F1M	A4M	P4M	P3M	P1M
F	1	0	0	0	0
A4	0	1	0	0	0
P4	0	0	1	0	0
P3	0	0	0	1	0
P1	0	0	0	0	1

FIGURE 2.8-6: Field-of-View Matrix

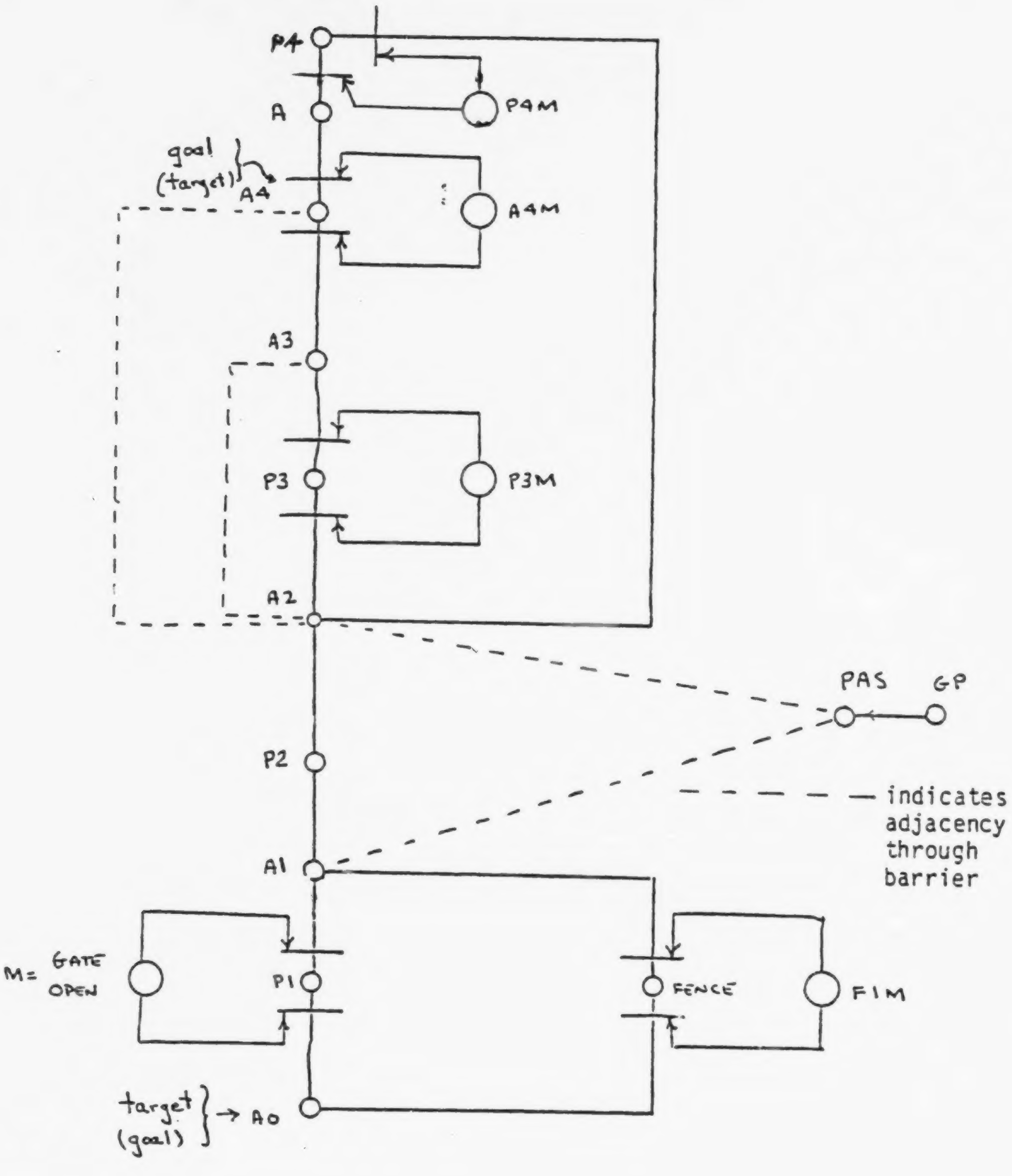
Other information is needed even in this simple case to do a complete analysis[20]. This includes:

(1) location of monitors
(2) monitor system (MC) interconnection
(3) location of MC elements
(4) utility (power) interconnections

To illustrate the analysis procedure, we would construct the Petri Net for this example although Boolean equations are actually used as input. In general, we would only construct the adjacency matrix, not draw the net. Figure 2.8-7 shows the Petri Net for this example. The rules for constructing the net are:

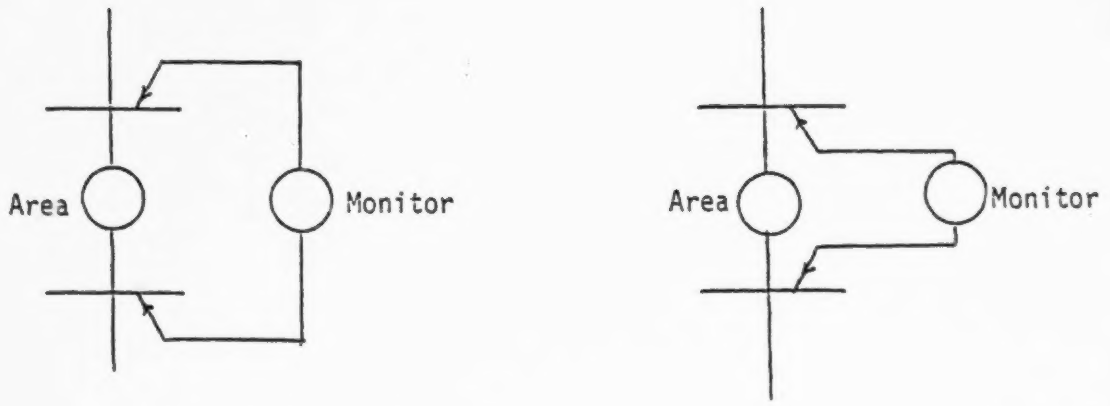
(1) draw the area adjacency graph (Figure 2.8-5)

(2) surround areas and portals which are monitored by AND gates (controlled transitions). This operation basically states that the transition into an area (or portal) is monitored or controlled. If the area were monitored for a required presence, then the arrows would be on the opposite side of the transition as Figure 2.8-8 shows.


P4
A
P4M
goal (target)
A4
A4M
A3
P3
P3M
A2
PAS
GP
P2
A1
M= GATE OPEN
P1
FENCE
F1M
target (goal)
A0

— — — — indicates adjacency through barrier

FIGURE 2.8-7: "Petri Net" For Simplicity Uranium Corporation


Area
Monitor
Area
Monitor

Area Monitored For
Unauthorized Personnel Presence

Area Monitored For
Authorized Personnel Absence

FIGURE 2.8-8: Area Monitoring

(3) connect the monitors to the information network by adding graphs from each monitor to the Petri Net. The MC adjacency matrix contains this information. This enhanced Petri Net is shown in Figure 2.8-9. The MC&A information flow adjacency matrix is shown in Figure 2.8-10.

The direction of the flow of information in Figure 2.8-9 appears to be reversed. This is because we are looking at the inhibition of the movement through the system in the Petri Net. The example could be resolved to much greater levels of detail, including MC&A element location, detection failures, utility flows, mode of process operation effects on monitors, etc., but first we will discuss the network shown in Figure 2.8-9.

The adjacency matrix for the Simplicity Uranium Corporation (Figure 2.8-11) was constructed from the rules described earlier in this paper and the Petri Net of Figure 2.8-9. In general, the drawing of the Net is not necessary and has been shown here only for clarity. The input to the reachability code may be either the adjacency information or a Set of Boolean Equations as described in Volume 2. This adjacency matrix represents 25 simultaneous Boolean equations which describe the facility and MC&A system. The corresponding reachability matrix has been solved using a modified version of the reachability code (CLAMOR) described in Section 2.8.1.3. The reachability element of interest is [r A0, A4], which addresses the question: Can the adversary travel from A0 to A4? The solutions found by the reachability code were:

```
r(A0), A4) = P3M A4M M   +
             P3M F1M A4M +
             P4M A4M M   +
             P4M F1M A4M
```

This equation indicates that the adversary cannot reach from the outside (A0) to the target area (A4) unless at least one of the terms in the equation is true. This means, for example, that if P3M and A4M and F1M are not operating, the adversary could win.

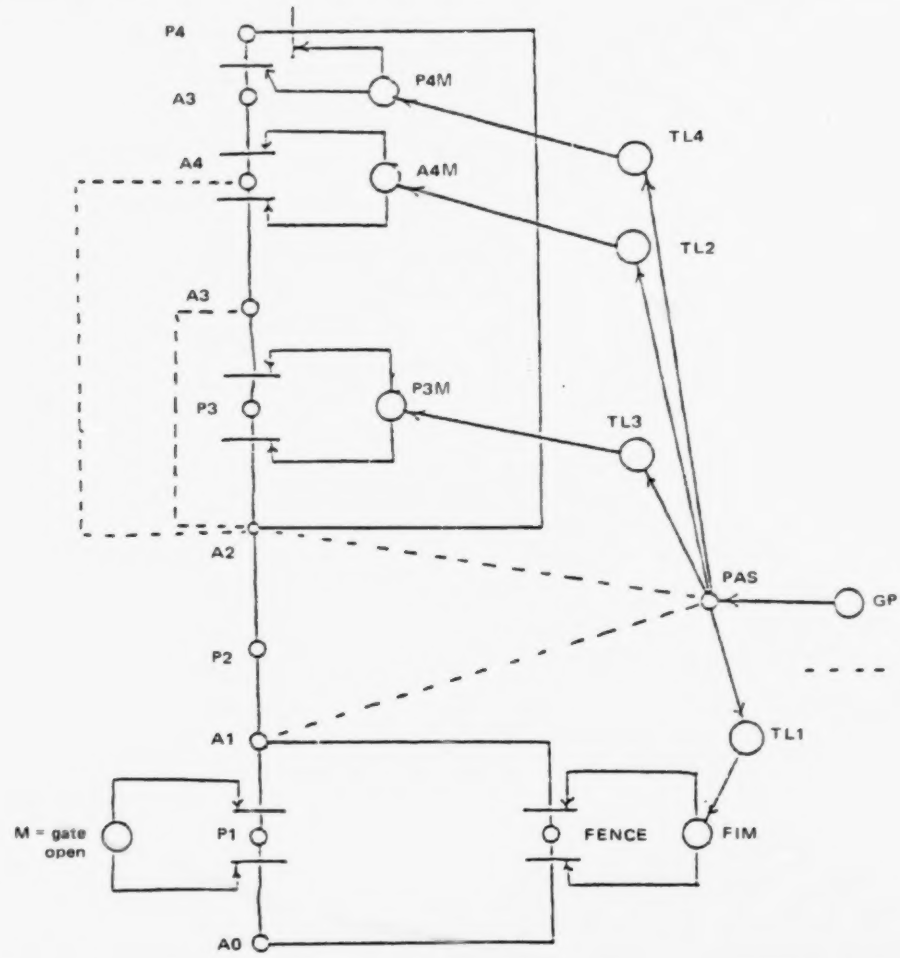

P4
A3
P4M
A4
A4M
TL4
TL2
A3
P3
P3M
TL3
A2
PAS
GP
P2
A1
TL1
M = gate open
P1
FENCE
FIM
A0

FIGURE 2.8-9: "Petri Net" for Simplicity Uranium Corporation With MC&A Information Flow Added

	A4M	P4M	P3M	F1M	TL1	TL2	TL3	TL4	PAS	GP
A4M						1				
P4M								1		
P3M							1			
F1M					1					
TL1									1	
TL2									1	
TL3									1	
TL4									1	
PAS										1
GP										

FIGURE 2.8-10: MC&A Information Flow Adjacency Matrix

	A1	P1	A2	P2	P3	A3	A4	P4	F	A	P4M	TL4	P3M	TL3	F1M	TL1	A4M	TL2	R1	R2	R3	R4	A0	PAS	GP
A1		M		1					F1M																
P1	1																						1		
A2				1	P3M			P4M																	
P2	1		1																						
P3			1			1																			
A3					P3M		A4M																		
A4						1				1															
P4			1							1															
F	1																						1		
A							A4M	P4M																	
P4M								A+ A2																	
TL4											1														
P3M					A3+ A2																				
TL3													1												
F1M									A1+ A0																
TL1															1										
A4M							A+ A3																		
TL2																	1								
R1																	1								
R2															1										
R3													1												
R4											1														
A0		M							F1M																
PAS												1		1		1		1							
GP																								1	

FIGURE 2.8-11: Adjacency Matrix For "Petri Net" of Figure 8

The equations can be investigated in other ways as well. For instance we could ask, if the Primary Alarm Station, PAS, were in collusion with the adversary, could he win? The reachability matrix yields the following information:

```
r[PAS, F1M] = 1
r[PAS, P3M] = 1
r[PAS, F1M] = 1
r[PAS, A4M] = 1
r[PAS, M]   = 0
```

Substituting into the equation for AO, A4 yields:

```
r[AO, A4] =  PAS PAS M   +
             PAS PAS PAS +
             PAS PAS M   +
             PAS PAS PAS
r[AO, A4] =  PAS
```

This says that if the PAS is in collusion with the adversary at AO, the adversary can win.

To determine the possibility of an adversary being successful by tampering, we modeled the locations of the MC&A elements. Figure 2.8-12 repeats the earlier example with one modification: the PAS is made adjacent and without a barrier to area A4. Then the adjacency matrix elements become

a[A4, PAS] = 1, and
a[PAS, A4] = 1

Solving the equation using the reachability code for element r[AO, A4] yields the same solution as before.

The interesting case is, of course, whether the adversary could reach AO. If he started inside at A4 with the modified adjacency matrix, we find that:

r[A4, AO] = 1

or the adversary <u>can</u> reach AO.

The example has been kept deliberately simple, in fact incomplete, to illustrate. Figure 2.9-13 gives a more accurate and detailed view of the example. In this figure we can see that the adversary has many opportunities to tamper. If he first knocks out the Electric Company [ELECT CO.], the fence monitor [FIM] fails. Since he has access to Area A2 he can disable the transmission lines to P3M, A4M, and thus reach from A0 to A4. If the reachability matrix for this case is solved, it will be found that

$$r[A0, A4] = 1$$

Data that describe the Petri net are generally input in the form of local Boolean equations which describe the interconnection of the networks. This was shown in Volume II.

2.8.1.3 Material Accounting System Analysis. The analysis technique for the Material Accounting System is similar to that described for the Material Control System[21]. The basic data flows are modeled along with unit models for each of the MA components. In addition, we used the Petri Net Transitional Element to model data and component protection. The basic modeling philosophy has been described in Volume II of this report and is briefly summarized in the series of Figures 2.8-14 through 2.8-18.

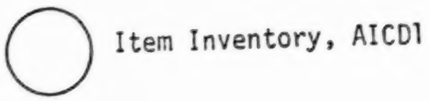


FIGURE 2.8-14: Identify Late Detection Systems

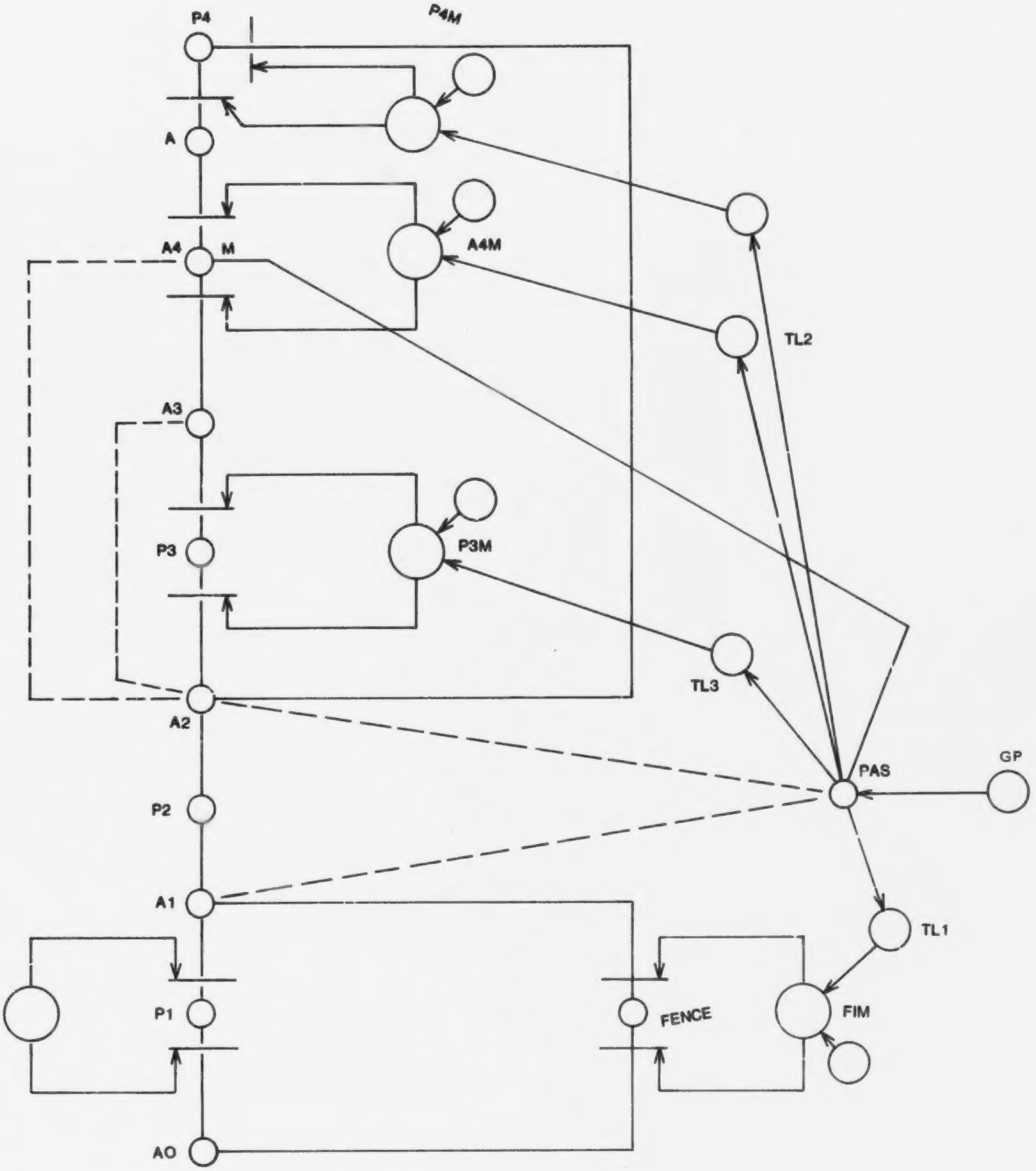

P4
P4M
A
A4
M
A4M
TL2
A3
P3
P3M
TL3
A2
PAS
GP
P2
A1
TL1
P1
FENCE
FIM
AO

Figure 2.8-12: "Petri Net" For Simplicity Uranium Corporation

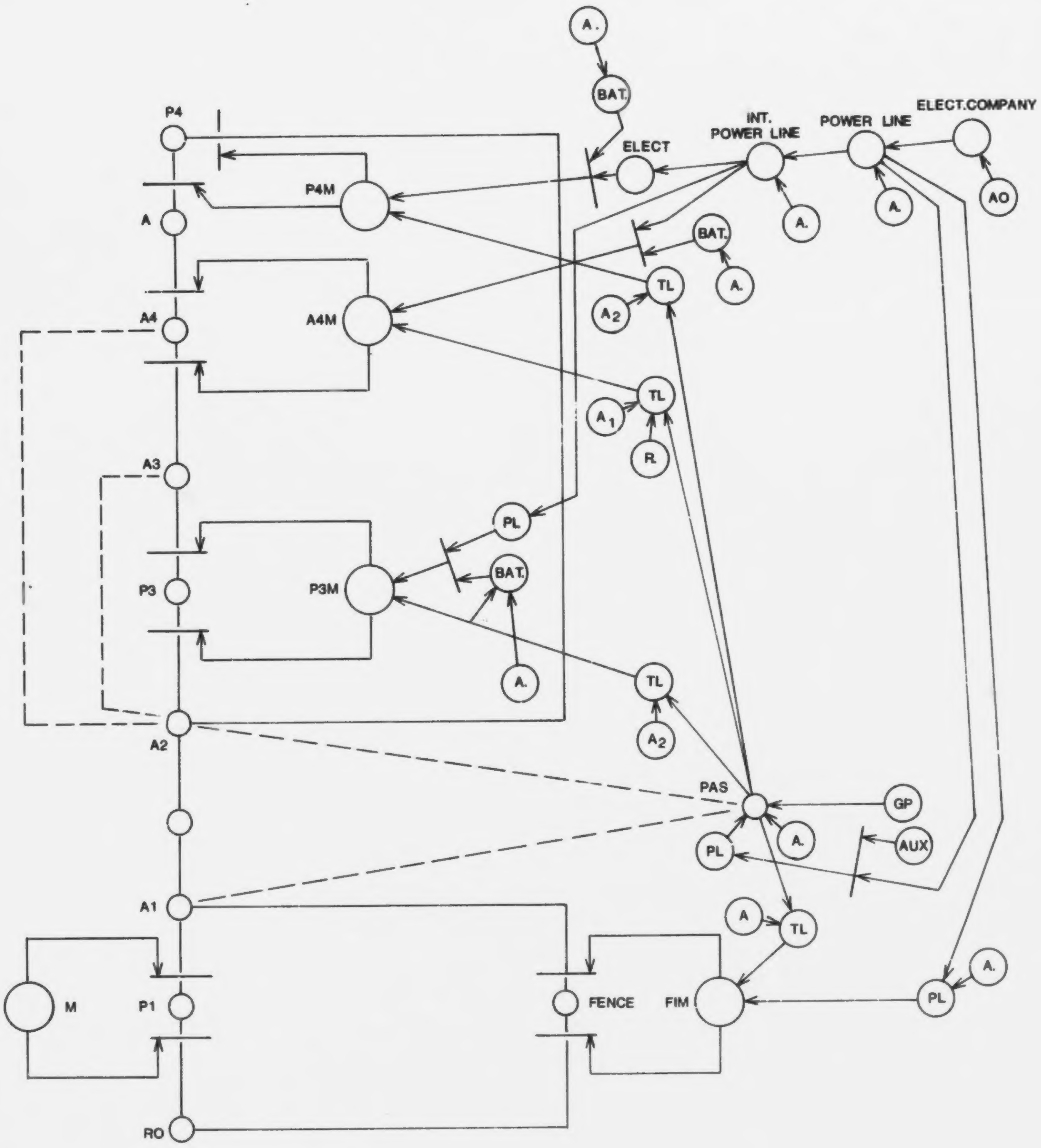
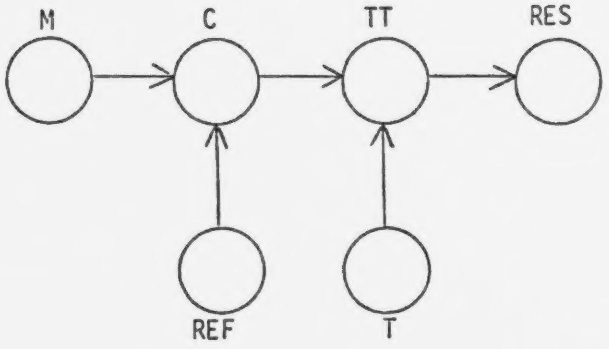
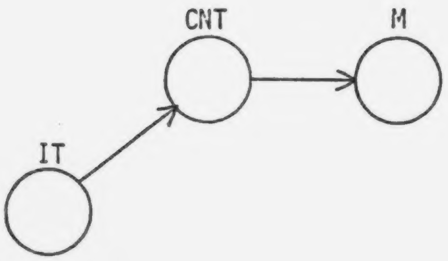

A.
BAT.
P4
INT. POWER LINE
POWER LINE
ELECT.COMPANY
ELECT
P4M
A
A.
A.
AO
BAT.
TL
A.
A_2
A4
A4M
TL
A_1
R.
A3
PL
P3
P3M
BAT.
A.
TL
A_2
A2
PAS
GP
PL
A.
AUX
A1
A
TL
A.
M
P1
FENCE
FIM
PL
RO

Figure 2.8-13: Higher Resolution Net For Example


M
C
TT
RES
REF
T

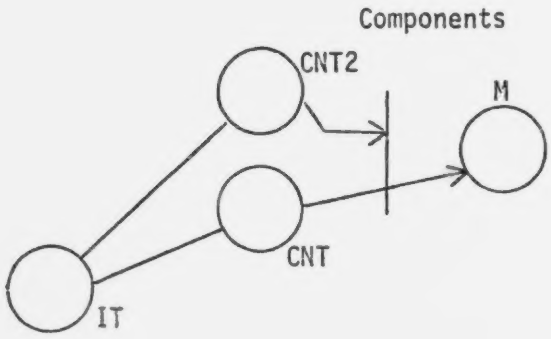
M = Measurement
REF = Reference
C = Comparison
TT = Threshold Test
T = Threshold
RES = Response

FIGURE 2.8-15: Identify Components of Detection System



IT = Items
CNT = Count

FIGURE 2.8-16: Identify Sources of Data for All Components


Components
CNT2
M
CNT
IT

CNT2 = Second Independent Count

FIGURE 2.8-17: Identify Protection For Each Data Flow

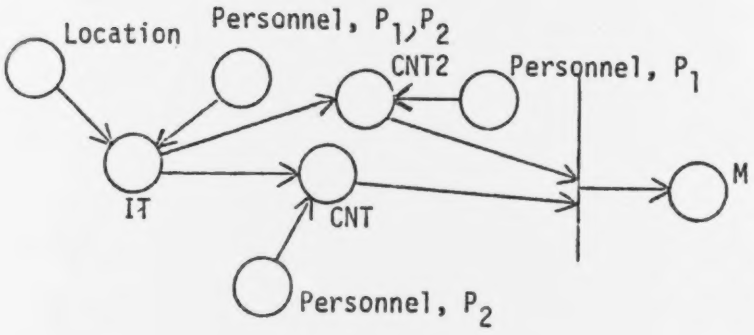


FIGURE 2.8-18: Identify Unit Models, Locations, Personnel

2.8.1.4 Boolean Analysis Code Description - CLAMOR. CLAMOR, the SAA Boolean Substitution Code, is designed to process the equations which describe both the Material Control and Material Accounting Systems. This section, based on Reference 22, provides a technical description of CLAMOR.

CLAMOR provides a computer-based tool of the Level 4 analysis Structured Assessment Approach. To facilitate the high modularity implicit in such a tool, CLAMOR adheres to a block-overlay programming structure. The code within various program blocks is easily extended or eliminated as the need arises. The resident overlay is merely a skeleton of basic system and input/output routines. These routines deal with the peculiarities of the host computer's operating system, shuffle the data blocks around core, and provide the dynamic storage allocation that allows CLAMOR to handle arbitrary problem sizes. All other sub-level code blocks are purely algorithmic. The coding associated with these blocks becomes memory resident only as the program flow requires it. A command language provides the user with an interactive method for controlling program flow.

CLAMOR's three major tasks roughly correspond to an input task, a computational task, and an output task. Initial adjacency information must be supplied to represent the particular network being modeled. Reachability is computed to expand this adjacency into a generalized set of paths through the network. The paths may then be presented as the principal output of the program, or the paths may be analyzed for reachability in terms of specific source nodes. Because data are currently kept in a redundant form, the user is free to invoke the sub-level tasks in any order. Subsequent sections will describe the tasks in detail.

CLAMOR is both an interactive and interrogative program. In the interactive portion of CLAMOR, the user issues a command that elicits some further action in response to a simple prompt. The prompt is usually indicative of which major task CLAMOR is considering. At this stage of talking to the program, the user is freed to choose commands, as long as the command is a legal option at the current level. The interrogative portion of CLAMOR follows. CLAMOR quizzes users to give additional information to complete the command, once a legal command is deciphered by CLAMOR. A simple example of this interaction, assuming that the user is executing CLAMOR from a teletype, follows:

```
•                   (CLAMOR prompts to receive a command)*
  FACTOR            (User enters a legal command)
       #            (CLAMOR prompts to receive a command at new level)
        NODE        (User enters a new command)
            From-I, To-J:  (CLAMOR begins interrogation)
                                1
                    To-J:
                            2
            From-I, To-J:
                            2 3
            From-I, To-J:
                            0
                    (CLAMOR prompts for a new command)
```

Given the opportunity, CLAMOR is very wordy. If the user has a general feeling for the tasks performed by CLAMOR, he can learn to manipulate the various commands by trial and error. The prompt and/or error messages should be self-explanatory. To avoid unnecessary delays in a time-sharing environment, the user may anticipate the questions CLAMOR will ask, and supply the needed information after the command. The following example illustrates this and accomplishes the same result as the preceding example.

```
FACTOR NODES 1 2 2 3 0
#
```

---

*Explanatory notes not appearing in the interaction are parenthetically inserted.

Notice that all input is space delimited. It may also be comma delimited. An *&* as the last item on an input line will allow your line to continue. Numeric values may be input as either integer or floating point. All responses requiring a YES or *NO* (the prompt will contain *?*) may be abbreviated. All commands may be abbreviated until they contain no more characters than necessary to make it distinct from another command. In practice, only the first four letters in a command are looked at. Also notice that the prompt, *.*, is changed to *#*, as CLAMOR changes tasks or levels. The usual method of exit from a level-task is to issue the prompt itself as a command or say *END* or *EXIT*.

CLAMOR processes separately each item in the message line, and when it reaches the end of the line, will either issue a prompt or continue interrogating. If an item in the message line is unexpected, CLAMOR will ignore the rest of the line, notify the user, and begin interrogating at the point it got confused. Since numeric and character strings are somewhat indistinguishable, there is a potential conflict in the way a message is handled. CLAMOR will never accept a character string when it expects a number, and vice-versa. The user can remedy an unexpected result from any input sequence by simply issuing the original command again. The user can give commands in any order at any given level-task. To imbed blanks in an item, or to force a numeric string to be interpreted as characters, enclose the input item with back-slashes, e.g., (\blank , \1234\).

It may seem cumbersome to respond to all the interrogations made by CLAMOR. In many cases, logic has been incorporated to provide short cuts. Often the user can avoid a series of interrogations by appropriately stringing commands together. Usually the user may anticipate the program and enter a new command whenever a null response is required to escape from a line of questioning. *YES* and *NO* options can often be sidestepped by: 1) immediately entering the data which would be requested if a *YES* answer were given; or 2) entering a new command instead of the *NO*. In all cases of such ambiguity the user may discover the right path only by trial and error. One of several recoverable error procedures may be initialized during the course of executing CLAMOR. Control will generally be passed to some neutral level of command interaction and hints provided to offer evasive action. These include expansion of scratch storage areas, re-examining input/output operations on disk transfers, etc. Certain program failures such as hardware errors or overflows which occur rarely cause the program to crash.

2.8.1.4.1 Controller Options. The usual method for executing CLAMOR is from the teletype or any controller simulating a teletype. The following useful commands may be issued at various levels:

```
SAVE filename
DISK filename
TTY
```

CLAMOR will ignore commands if the file name is blank or nill. The SAVE command will take subsequent message lines sent by the controller and save it in a disk file for later user perusal. CLAMOR will request a file name for creating this SAVE file. Given the DISK command, CLAMOR will ask for a file name, and all subsequent commands and command responses will be read from this file. When the file is exhausted, control will return to the controller. The disk file specified may be one created by a prior execution of CLAMOR in which the SAVE command was invoked. The TTY command, usually imbedded in a disk file of commands, will also return interactivity to the controller. If it becomes apparent that CLAMOR is caught in some kind of loop, or is reading a disk file incorrectly, the user may issue the control-E-I interrupt to regain control.

2.8.1.4.2 Data Management. The core storage requirement for each level-task is dynamically allocated as the user passes to that particular task. In general, only the basic data needed to define a model retains its integrity when alternating among the major tasks, ADJACENT, REACH, and FACTOR. Where a variable area of scratch storage is required, appropriate commands are provided to expand or contract this space from the default allocation. These commands will be discussed in the appropriate sections to follow.

2.8.1.4.3 Data Structure. The data which define a network model for CLAMOR consist of a basic adjacency matrix, $A=a_{ij}$, and a list of node labels. The labels have a one-to-one correspondence to the rows (or columns) of the adjacency matrix. An entry, a_{ij}, is non-zero in the adjacency matrix if a directed or conditional path exists from row-node-i to column-node-i. Details of the data arrangement in computer memory is divulged in the source listing of the code. Currently the data are memory resident, but appropriate modules may be rewritten to allow the data to reside on any supplemental storage device such as a disk.

2.8.1.4.4 Getting Help. The user can ask assistance from CLAMOR by issuing the following command:

HELP

CLAMOR will print a list of legal commands appropriate to the current level of operation at the teletype. If CLAMOR misunderstands a command, it will echo the command and ask if the user desires help. If the response is YES, it prints a legal set of commands. If the answer is NO or some legal command, perhaps followed by options, CLAMOR will simply proceed. In either case, the remaining items in the original list of messages containing the conflict will be purged.

Aside from any passing mention, the particular algorithms used within CLAMOR will not be discussed here since adequate documentation of them exists within the source listing. It should be mentioned that the programming language used is FORTRAN-77, and the particular machine and operating system on which development took place was LLNL's CDC-7600. Pertinent photostore directories are:

740468:PETRI:CLAMLIBS	(source listing)
:CLAMBINS	(binary modules)
:CLAMOR	(controllee)

2.8.1.4.5 CLAMOR Initialization. CLAMOR begins execution at the . prompt level. This is the main transfer point for entering or exiting any of the major level-tasks. Commands may begin on the execute line, if under teletype control, or at the first issue of a prompt. The user may issue commands appropriate to a new level immediately after commanding to transfer levels; he does not have to wait for the new level's prompt key.

The commands

```
DISK
TTY
SAVE
HELP
```

have been discussed. They will leave control at this level. To enter one of the major level-tasks, the user must issue one of the following commands:

```
ADJACENT    ZREACH    COMPLEMENT    DUMP
REDUCE      ZDEBUG    NOCYCLE       SOLVE
REACH       ZFREE     NOSOURCE      unary
FACTOR      ZPRINT    RELEVANT
```

Since the level-tasks are essentially independent, the user may immediately refer to the appropriate section to follow. If any model data are generated in the level-task chosen, this data is considered core resident and will follow the user around to another level-task. To leave the . level, and thus terminate CLAMOR, the user must issue one of the following commands:

```
    (period)
END
EXIT
```

In general, a control-E-I interrupt encountered at any level task will return control to the principal entry of that level task.

2.8.1.4.6 Output Information. Any program output is delivered to the teletype when CLAMOR is operating in the default mode. The commands

```
OUTPUT filename
TITLE
```

will redirect this output. CLAMOR will prompt for the filename if it is not on the command line. By specifying a non-blank filename, a disk file will be created which will receive any output generated within CLAMOR. The name may be the same as that appearing on any SAVE command. The type of output directed to either a SAVE-file or an OUTPUT-file differs. The SAVE-file merely serves to echo any messages received by CLAMOR. The OUTPUT-file serves to collect any messages sent by CLAMOR. If TTY is specified for filename, the current file will be closed and output redirected to the teletype. TITLE provides a method for inserting a user title into the output file. It will be automatically issued after an OUTPUT command. Should an output file become filled, a familied file will be created.

To monitor computer usage, use the command

```
TIME
```

which will print a table at the teletype. This table contains timing information for the various level-tasks as well as an overall resource usage.

2.8.1.4.7 Enlarging the Work Space. While processing data or computing reachability, CLAMOR may occasionally exceed the bounds of its work space. The user may issue the command

```
EXPAND freesize, swordlen
```

to dynamically expand storage areas. If the values of freesize or swordlen are omitted, CLAMOR will prompt for them. The first value, freesize, is used as the dimension of a large LCM buffer area used to store adjacency and reachability information. This buffer is normally expanded automatically as the need arises. However, if the buffer is exceeded generally by a prior inability to expand, CLAMOR will warn the user that it needs more free. All processing will stop so that the user can take evasive action, usually by employing the EXPAND. The second value, swordlen, is used to control the sizes of several SCM scratch vectors. It roughly corresponds to the number of words required to represent a single entry a_{ij} in the adjacency matrix or the path developed for a particular $< i,j >$ transition. Currently, swordlen is not expanded automatically. If either freesize or swordlen is input as 0 or null, the last previous non-zero value is used. Default sizes are 5000 and 100 respectively.

2.8.1.4.8 Adjacency Specification. A network model may be completely specified by an adjacency matrix, $A=a_{ij}$, whose entries represent a conditional path extending from node-i to node-j. The very term adjacency implies that a_{ij} is non-zero only for neighboring nodes in the network although in the general sense this need not be true.

The matrix is typically asymmetric and sparse. The nodes of the network are conceptually given some alphanumeric label and are in one-to-one correspondence to the row or column indices of A. Since the entries, a_{ij} are actually Boolean expressions, the adjacency information for the network becomes a set of Boolean equations:

$$\text{node}_j - \text{node}_i * [\ a_{ij}\ ]$$

If $a_{ij}=1$, then an unconditional, directed link exists from node_i to node_j. If $a_{ij}=0$, no link exists. In all other cases, a_{ij} represents a set of conditions which enable the link to exist. For example,

$$\text{node}_j = \text{node}_i * [\text{node}_k * \text{node}_1 + \text{node}_m]$$

means that a link exists from node_i to node_j provided the conditions represented by: a) node_k and node_i OR b) node_m are also met. Reference 16 develops these ideas more fully.

CLAMOR accepts extended adjacency information also, such as two nodes on the same path separated by a node in between.

In general, a network for CLAMOR is defined by a system of equations of the form:

$node_1 = DNF_1(node_1,\ldots,node_n);$

$node_2 = DNF_2(node_1,111,node_n);$

.

.

.

.

$node_n = DNF_n(node_1,\ldots,node_n);$

.

,

Each DNF_i is a disjunctive normal form or a simple sum of products involving the indicated arguments. Each $node_i$ is a user-defined, alphanumeric label for a node of the network. Each "i" is required. To the left of the "=" is a single node label called the terminus or "to-node". Any node appearing to the right of "=" is called a source or "from-node". When a user specifies ";" in place of a "to-node," he ends the input phase. It is convenient to think of the "=" as a "<=" so that each DNF_i represents a directed adjacency to $node_i$. For example,

A = B + C*D ;

is, in terms of the previous discussion, equivalent to the system

A = B *[1]

or

A = C *[D]

or

A = D *[C]

That is, there is a directed path from B to A, a conditional path from C to A, and a conditional path from D to A. CLAMOR processes each equation in the input stream to develop a self-consistent adjacency matrix. This means that any string of products is broken into a series of conditional paths ending on the terminus and starting from each factor in the product with the appropriate remaining factors as conditions. As CLAMOR encounters each new node label, CLAMOR adds it to the internal list of all node labels, and adds a new row and column to the adjacency matrix, A. If the user is forced to truncate the equation list, complemented nodes, which may appear within DNF_i, will give rise to equations looking something like

```
A = B*-C + -D*E ;
```

The strange pairing of operators is hard to look at, but it is necessary format for including a complemented event. If a to-node is duplicated, the right side of the equation is merged with existing expressions.

Entering the initial adjacency information defining the network for the model is simple. At the . prompt level, the user will issue the command

```
ADJACENCY  n
```

where n is the number of nodes in the network. At this command CLAMOR transfers to a sublevel which will begin interrogating him.

If the user waits to be prompted for input during the ADJACENT command, he will get a series of interrogations that basically ask for phrases from which to build a final equation. The to-node is asked for first, then the =, then a series of from-node - operator pairs and perhaps a terminating ;. If the user is familiar with the simpler and preferred input approach, he will be able to respond correctly if caught in the interrogation mode.

Once CLAMOR has an adjacency matrix specification, it encodes each node label with a unique prime number, and represents each condition a_{ij}, as a sum of products of primes. Thus, subsequent manipulation of the adjacency becomes a numerical rather than symbolic problem.

2.8.1.4.9 Path Generation. Once the adjacency information for the network has been specified, the problem becomes one of identifying paths between various non-adjacent nodes in the network that reads: We have extended the Warshall algorithm [7] for computing the transitive closure, A*, of the adjacency, A, to manipulate the conditional adjacency implied by the Petri formalism.

We chose this graphic, analytic approach over the process simulation approach, because current needs require examining all paths between all combinations of terminal nodes. The algorithm is deterministic. It takes a finite amount of time and automatically guarantees completeness. No extra storage is required. The transitive closure of an adjacency matrix provides information about the global connectivity of a network originally specified by conditional local connectivity. The principal output of the algorithm is a set of Boolean equations or paths

$$\text{node}_j = \text{node}_i * [\ a^*_{ij}\ ]$$

where node_j is said to be "reachable" from node_i provided conditions a^*_{ij} are satisfied. In general, the conditions are complicated Boolean expressions involving other nodes in the network. If the initial adjacency, A, contains any loops (non-zero diagonal elements), the algorithm will condense the information and lose some detail, although A* will be complete in the Boolean sense. The authors feel that such loops will not exist in the current application.

While at the _._ prompt level, the simple command

REACH

will transfer control to the sub-levels required to compute the transitive closure of the adjacency. If all goes well, CLAMOR will return control to the _._ prompt level upon completion of this task. Should the calculation be interrupted because CLAMOR does not have enough scratch space, then the user can issue the ZREACH command (see the section on Miscellaneous Z-Commands), and continue the calculation after taking evasive action, such as issuing the EXPAND command.

2.8.1.4.10 Elimination Options. Since prime numbers are used to encode the initial set of Boolean expressions represented by the adjacency, A, the closure generated contains sums of products of primes. These products may extend over several word boundaries. Several elimination options have been incorporated into CLAMOR that provide shortcuts into the Warshall algorithm and help to minimize the number of addends in a particular a^*_{ij}. Under user control, minimal paths, dominant paths, loopless paths and/or unary paths can be selected. Clever use of Euclid's Theorem allows these various path reduction options to be done on the fly.

Issued at the . prompt level, the commands

```
COMPLEMENT answer
RELEVANT answer
NOSOURCE answer
NOCYCLE answer
UNARY answer
```

will have an effect on any subsequent REACH command. If the use omits the answer, CLAMOR prompts for it. If the user's answer is OFF, that elimination option will be turned off. Any other response, including, perhaps, another command, will turn that elimination option on, and it will remain in that state until explicitly turned off.

The effect of each elimination option is shown in the following example. Consider a conditional path obtained when all options are in an initial OFF state:

```
T = S * [ 1 + A + A*B*A + A*B*A*C + S*G + (-S)*H ]
```

The quantity in brackets is essentially what is computed by invoking the REACH command, and represents the path from source node, S, to terminal node, T. If option NOSOURCE is turned ON, this quantity would be computed as

```
[ 1 + A + A*B*A + A*B*A*C + G + (-S)*H ]
```

The redundant factor, S, is removed from the fifth term. COMPLEMENT will similarly eliminate the inconsistent final term:

```
[ 1 + A + A*B*A + A*B*A*C + G ]
```

When the option NOCYCLE is turned on, all cycled paths are eliminated:

[1 + A + A*B + A*B*C + G]

That is, the expression is reduced to first order in any given term.

The RELEVANT option will cause only the minimal expressions to be retained:

[1 + A + G]

Interpreting the bracketed expression as a condition justifies this reduction, because the single condition A makes conditions A*B and A*B*C redundant.

The UNARY option is merely a special case of RELEVANT and will further reduce the condition to:

[1]

For most applications it is suggested that all elimination options be turned on. This is the default condition. The essential information for each path is retained and the economy in storage is significant.

2.8.1.4.11 Saving the Reachability Calculation. CLAMOR takes adjacency information and processes it into a transitive closure A*, at the REACH command. Other codes and CLAMOR's postprocessors, such as FACTOR and SOLVE can use this information.

To store the data rather than carry it on the core, at the . level, the user commands

```
DUMP filename
```

which will deposit all pertinent information onto the disk file, filename. If the user omits filename, CLAMOR will prompt for it. Currently, it is suggested that this command be used before using FACTOR or SOLVE. The entire A* (or A) arrays will be sent to disk along with essential control information.

2.8.1.4.12 Generating Output. To obtain printed results of the reachability calculation, the user may issue the level . command

```
FACTOR
```

A new set of # level commands are available as a result. Some of these,

```
OUTPUT
HELP
END
EXIT
#
```

have already been discussed. The last three return the user to the . prompt level.

If a previous DUMP command was issued, the user may recover data from disk by using the level # command

```
READ filename
```

where filename is the name of that file obtained when DUMP was used. If the user omits filename, CLAMOR will ask for it. If the READ command is not invoked, processing will continue from any data that are core resident. Because of the significant savings in core space, we suggested that the READ command be used for large (greater than 50 nodes) networks.

To obtain the printed version of the paths between all combinations of terminal nodes in the network, simply issue the level # command

```
ALL
```

Unless the network is quite small, the user will probably have previously issued an OUTPUT command to avoid a lengthy (one-sided!) teletype conversation.

To select particular paths, the user may command

```
NODE node_i, node_j
```

to print the single path from $node_i$ to $node_j$. This command is recursive and will continue to prompt for node pairs until a /blank/ or carriage return is encountered for one of $node_i$ or $node_j$. If the keyword ALL is used for $node_i$, a list of all paths which terminate on $node_j$, then all paths originating from $node_i$ will be printed.

When any of these output commands are encountered, the prime-encoded Boolean expression contained in the reachability matrix is reconverted to a symbolic representation. In the default mode, the Boolean expression printed for each path will not be factored. If level # commands

```
ZLONG          (true by default)
ZSHORT         (false by default)
```

are issued, the true/false status of each is complemented. When the status of ZSHORT is true, a factored expression is produced. When the status of ZLONG is true, an unfactored expression is produced. (If both ZSHORT and ZLONG are false, no output is produced.) Each time the user enters FACTOR, the status of the flags is returned to their default condition. A word of warning, the algorithm for ZSHORT is very inefficient. The user is encouraged to make use of the SOLVE command whenever possible.

2.8.1.4.13 Solving The System The FACTOR command previously discussed has a limited use in that it merely presents the paths developed by the reachability calculation or perhaps those paths used as input during ADJACENT. Very often the paths (or equations relating a particular source to a particular sink node) are lengthy and contain conditionals that with suitable substitution would result in a simpler expression. To obtain an equation describing the reachability of a specific terminus node in terms of a set of source nodes, the level . command

```
SOLVE filename
```

may be invoked. If the user omits filename, CLAMOR will prompt for it. Filename should be the name of a disk file previously created with a DUMP command. If filename is non-blank, all resident data will be purged, and data for the solution process will be drawn from the disk file. The user should be warned that core-resident data generated elsewhere in CLAMOR are not compatible with SOLVE and vice-versa. (The basic difference is the arrangement of key arrays in storage. SOLVE works with non-square arrays.) SOLVE will only proceed if it is allowed to read a disk file. The data that become core-resident as a result of SOLVE reading a file will in general not be compatible with other sub-level tasks. (This restriction may change in the future.)

The purpose of sub-level task SOLVE is to accept data from the global reachability or transitive closure, A*, and perform a systematic substitution process that reduces all conditions in A* to expressions involving a few user-specified source nodes. After the initial SOLVE command, the user is interrogated in the following manner:

<source node> node_1
<source node> node_2
.
.
.
<source node> node_m
<source node> /blank/ or carriage return
<terminus node> tonode

Here the user is recursively asked for a set of source nodes, (node_1, node_2,...,node_m), and a single terminus node, tonode. SOLVE will then generate the system of all equations from A* with any node_i as a source node. In particular, it will construct the equation

$$\text{tonode} = \text{DSN}(\text{node}_1,\ldots,\text{node}_m\text{m other-nodes})$$

which is the disjunctive normal form expressing the reachability of tonode from any of the source nodes. The other-nodes arise from the various conditionals present in A*. SOLVE will eliminate all the other-nodes from this expression by examining their reachability from the original source nodes. The result will be a single disjunctive normal form involving only source nodes:

$$\text{tonode} = \text{DSN}(\text{node}_1,\ldots,\text{node}_m)$$

In Petri terminology, the disjunctive normal form obtained by SOLVE provides all combinations of source nodes, which, if marked by a token, would result in tonode being potentially reachable. In other words, all the local conditional information used in constructing the adjacency is reduced to conditionals involving only source nodes. In the ideal case, all the source nodes would be independent, that is, they would be non-reachable from one another. If this is not the case, SOLVE will first find expressions for the dependent source nodes in terms of the independent nodes and the resultant equation for tonode will also be in terms of the originally dependent nodes. If there is no independent node at the start, or if any of the user-specified dependent source nodes cannot have their conditionals reduced to expressions involving only other source nodes, then SOLVE will treat these nodes independently. The user will be alerted to any situation involving dependent source nodes. If an equation for tonode cannot be obtained then SOLVE concludes that tonode is isolated from the designated source nodes. That is, there exists at least one conditional that could not be expressed in terms of source nodes alone. If an equation is found, SOLVE prints it out. If no independent source nodes were found, it is possible that a subset of the original set of source nodes would also generate a reaching path to tonode. It should be mentioned that complemented nodes (either of source nodes or of other-nodes) may appear in the final expression for tonode. These are interpreted merely as hints as to the choice of paths taken when alternate paths appear in the conditional. During the computation of A*, and during SOLVE, the appearance of isolated complemented nodes in the final expression is not inconsistent with the overall reachability.

2.8.1.4.14 Reducing The Adjacency Information. The user can abbreviate the initial adjacency information by various condensation, pre-substitution, and partitioning methods. This will streamline the reachability calculations and simplify the factoring problem. Coding associated with the REDUCE module of CLAMOR is still in development and will not be discussed here.

2.8.1.4.15 Miscellaneous Z-Commands. The Z-commands within CLAMOR are hidden commands in that they will never appear in a list produced by HELP. They are basically debug tools. The commands

```
ZPRINT
ADEBUG
```

issued at the . prompt level cause additional diagnostic printout of little use except to the author. The command

```
ZFREE
```

will print out the amount of scratch storage actually used by the various sub-level modules, and is helpful in deciding appropriate values for the EXPAND command. The command

```
ZREACH
```

will restart the reachability calculation from where it left off if the user were forced to make use of the EXPAND command. That is, if the REACH command returned with an error condition (usually that the scratch area swordlen was too small) the user may simply continue the calculation after taking evasive action.

2.8.1.4.16 An Example. Figure 2.8-19 shows a typical network with conditional adjacency imposed by a Petri net.

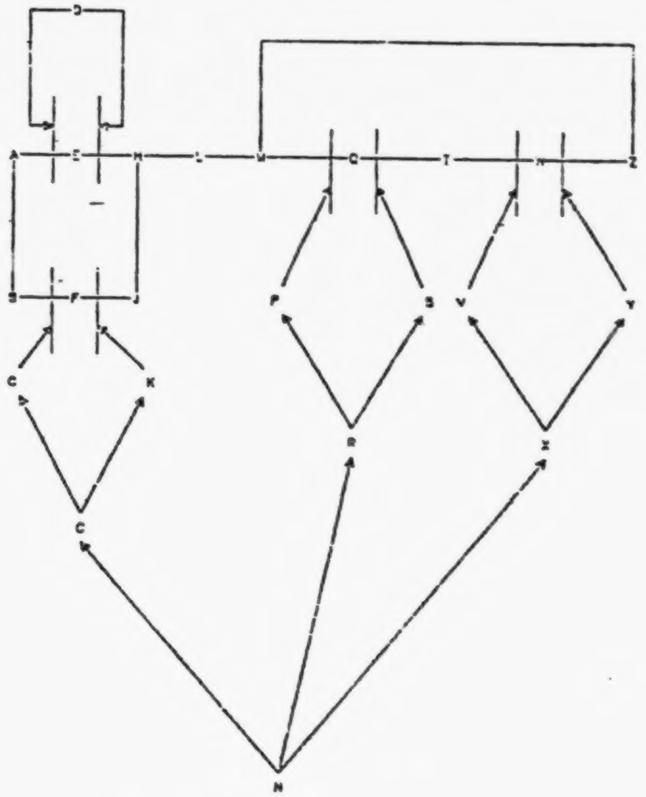


FIGURE 2.8-19: A Sample Network

A disk file, EXAMPLE, was previously created with CLAMOR by using the SAVE command. This file, containing the adjacency information for the network in Figure 1, looks like:

```
adjacent 23
a=b+e; b=o*f; c=g; e=o*d+d*h; f=b*c+j*k; g=n; h=e+j+l; &
j=f+h; k=g; l=h+m; m=l+z+a; p=r; q=m*p+t*s; r=n; s=r; &
t=q+w; v=x; w=t*v+z*y; x=n; y=x; z=w+m; &
;
```

A subsequent execution of CLAMOR to perform the reachability calculation on EXAMPLE and to print (in an OUTPUT file) a list of paths between all nodes is accomplished by the following dialogue. The user also created a dump file for later use. Note the switch to TTY output.

```
CLAMOR output examount disk example /  .3 .2
.TITLE BLOCK FOR OUTPUT:
this an example from figure 1.
.unary off complement off reach
.factor zlong zshort all
     #out tty
     #node a t n t / /

(18  = 1)
t            =o   *     (c+d)*(p+y)

(18  = 11)
t            =n   *     (b+j)*(p+y)+m+z+t

     #     #
.dump forsolving     .

ALL DONE
```

In this example the user was interested in the <a,t> and <n,t> transitions. Note the optional conditions required when traversing the network from node a to node t. For certain applications it may be necessary to expand the bracketed quantities in terms of certain designated source nodes. This can be done by manually substituting for the terms within the brackets from other equations present in examount. Employing the SOLVE command, the user can find that, in the Petri sense, it is sufficient to place a token at nodes a and n - the reachability to node t is assured. That is all the other conditionals can be expressed in terms of a and/or n alone. However, it is not possible to reduce the conditionals in terms of only a. The following example shows how the SOLVE command was used to eliminate all conditionals except a or n for the reachability t. Note the use of the previously created dump file, for solving. Also note in the output generated that a dependent source node was found. The second equation for node a is that which was used in the substitution process.

```
CLAMOR /  .3 .2
.solve forsolving
SOURCE NODE SYMBOLS(S): a n
TERMINUS NODE SYMBOL:   t

SUBSTITUTION PROCESS

INDEPENDENT SOURCE NODES:

     n = 1   *[ (n) ]
```

```
DEPENDENT SOURCE NODES:
     a = 1   * [ n*(  (b+j) + (d+k)*(m+z+t)   ) ]
     a = 1*  [ (a) ]
TERMINUS SUBSTITUTION:
     t = 1 *[ a*n ]
.solve forsolving a     t
SUBSTITUTION PROCESS
INDEPENDENT SOURCE NODES:
     a = 1 *[  (a) ]
TERMINUS SUBSTITUTION:
===NO PATH CONNECTS *SOURCE* TO        t
.end
ALL DONE
```

Following is a sample of the output file, examount.

```
this is an example from figure 1.
elimination flags used by -reach-:
unary=.false.     nocycle=.true.     relevant=.true.
nosource=.true.     cmplement=.false.

(2 <= 1)
b         =a             *[  1   ]

(3< = 1)
e         =a             *[  d   ]

(4 <= 1)
f         =a             *[  x+k*s  ]

(8 <= 1)
h         =a             *[  c+d  ]

(9 <= 1)
j         =a             *[  c+d   ]

(12 <= 1)
1         =a             *[  c+d   ]

(13 <= 1)
m         =a             *[  c+d   ]
```

A complete CLAMOR input listing for a Material Accounting detection system along with the solved input can be found in Volume III of this document. A derivation of this input is given in Volume II of this document. The second page of the listing contains the symbol map used in the adjacency matrix description of the equations. Below this are the results of solving the input equations. For example, the first solved equation for AICD1 indicates that:

> between Bimonthly Inventories (ABM) the system (AICD1) can be defeated by personnel with access to locations L9 or L12.

2.8.2 The CLAMOR code listing is found on the microfiche labeled SAA-Level 4.

ACKNOWLEDGMENTS

This volume represents an intensive research and development effort by many individuals. In particular the authors would like to thank Dragoslav Siljak of the University of Santa Clara for his consulting services on this project and T. R. Rice and S. L. Derby of Applied Decision Analysis who participated in the design of the analysis techniques.

The authors wish to thank A. D. Scarpetti and N. M. Sanford for editorial assistance and S. K. Marting, G. M. Simpson, and D. M. Zerr for preparing the manuscript.

A. A. Parziale
I. J. Sacks
D. W. Freeman
C. J. Patenaude
P. A. Renard
D. J. Ross

BLANK PAGE

REFERENCES

1. D. J. Ross, Input Specification for Level 1-3 Pathfile Builder, Lawrence Livermore Laboratory, Livermore, CA., MC 79-09-05-M (1979).

2. D. Siljak, C. J. Patenaude, A. A. Parziale, Processing Large Monitored Graphs, Lawrence Livermore Laboratory, Livermore, CA., MC 79-242-D (1979).

3. P. A. Renard, "Warshall on Symmetric Matrices", LLNL, MC 79-207, February 15, 1979.

4. I. J. Sacks, Extension to Level 1.5/3.5--Weak Collusion Tests, Lawrence Livermore National Laboratory, Livermore, CA., MC 79-08-06-D (1979).

5. D. J. Ross, Expanding Access and Control Lists for Facility Personnel, Lawrence Livermore National Laboratory, Livermore, CA., MC 79-09-12-M (1979).

6. I. J. Sacks, A. A. Parziale, M. Schrot, and J. Long, A Structural Approach to the Assessment of Material Control and Accounting Systems, LLNL, February, 1978.

7. Tremblay, Sorenson, An Introduction to Data Structures with Applications, P389, McGraw Hill, 1976.

8. I. J. Sacks, C. J. Patenaude, "Level 1.5--Weak Collusion Test for the Structured Assessment Approach", LLNL, MC 79-338, July 6, 1979.

9. "The MAIT Method for Analysis of Facility Safeguard Against Insider Collusion--User's Manual", NUREG/CR-0532, SA178-960-LJ, Vol. 2. Available only from Science Applications, Incorporated, LaJolla, CA 92037.

10. I. J. Sacks, C. J. Patenaude, "Interim Level 3 Analysis", LLNL, MC 78-1125-D, May 30, 1979.

11. C. J. Patenaude, I. J. Sacks, "Revised Level II/III Flowcharts", LLNL, MC 79-296-D, May 30, 1979.

12. T. Rice, S. Derby, Characterization of Analytical Procedures to Calculate the Probability of Successful Diversion of SNM, September 25, 1978, Applied Decision Analysis, LLL internal memo MC 78-1037.

13. S. Derby, Implications from Two Definitions of the Level 3 Criterion, September 15, 1978, Applied Decision Analysis, LLNL internal memo MC 78-1034.

14. C. J. Patenaude, "Bounding on the Probability Code", LLNL, MC 79-364-D, July 20, 1979.

15. A. A. Parziale, "Safeguard System False Alarm Rate", LLNL, MC-358-D, July 13, 1979.

16. I. J. Sacks, "Level 3.5 Second Phase Collusion Test", LLNL, MC 79-313-D, June 14, 1979.

17. I. J. Sacks, "Techniques for the Determination of Potential Adversary Success With Tampering (Level 4.1)", LLNL, MC 78-928-D, October 17, 1978.

18. A. A. Parziale, "Modeling Adversary Tampering of a Safeguard System With a Petri Net", LLNL, MC 78-514, June 30, 1978.

19. I. J. Sacks, A. A. Parziale, Unit Models, LLNL, MC 78-884, September 10, 1978.

20. I. J. Sacks, A. A. Parziale, M. Schrot, and J. Long, A Structural Approach to the Assessment of Material Control and Accounting System, LLNL, February, 1978.

21. A. A. Parziale, I. J. Sacks, T. Rice, and S. Derby, The Structured Assessment of Facility X, NUREG/CR-0791, UCRL-52765, January 8, 1979.

22. P. A. Renard, "CLAMOR Rev.", LLNL, MC 79-96, September 6, 1979.

APPENDIX A

GUIDANCE TO DOCUMENTATION

This section is intended to provide guidance to the reader who wishes to understand in detail the concepts embodied in the Structured Assessment Approach.

Documentation for the Structured Assessment Approach has been generated as a continuing process. In general, technical memoranda were written describing the concepts of each portion of the SAA followed by computer code development. In addition to the technical documentation, material has been generated about the implications of results of SAA processing.

Table 1.3-1 outlines the relevance of each of the documents relative to:

SAA Licensee submittal data (LSD) input
SAA Preprocessor operation
SAA Level 1 analysis
SAA Level 1.5 analysis
SAA Level 2 analysis
SAA Level 3 analysis
SAA Level 3.5 analysis
SAA Level 4 analysis
SAA performance measure outputs
Development of Modeling Rationale
Computational considerations (algorithms)
Management/procedural considerations

The following pages contain an abstract of each of the documents. The reader must keep in mind that more recent documents display current understanding and coding where past documents contain the seeds of further development. The sequence of documentation is important in that it reveals some of the motivations in SAA development.

To further assist the reader, it is recommended that the documents be read in the following order:

DOCUMENT #

34
1
3
4
8
25
38

ANNOTATED BIBLIOGRAPHY

1. Parziale, A. A., Sacks, I. J., Rice, T., and Derby, S., "The Structured Assessment of Facility X", LLL MC 79-12-D, Volumes I and II, January 8, 1979.

   This report describes the application of the Structured Assessment Approach (SAA) to Facility X, a full-sized facility. Volume I of the report is an executive summary which describes overall conclusions and results for each level of the SAA. Volume II goes into the details of Licensee Submittal Document input data (LSD) and the complete analysis.

2. Sacks, I. J., Parziale, A. A., and Renard, P., "Modeling of Procedures", LLL, MC 78-1374-D, December 18, 1978. (Draft)

   This memo describes our approach to modeling procedures which are a part of a material control and accounting system.

3. Rice, T. and Derby, S., "Overview of the Structured Assessment Approach and Documentation of Algorithms to Compute the Probability of Detection", Applied Decision Analysis (ADA), LLL, UCRL 13937, December 15, 1978.

   This major report written by ADA provides an SAA description, as well as the rationale behind the SAA and the leveling of analyses. ADA's probability calculation algorithms are also discussed in detail. A good conditional probability description of the SAA is given.

4. Sacks, I. J. and Renard, P., "ASAP - Accounting System Analysis Procedure", LLL, December 6, 1978. (Not available at this time.)

   This paper will describe the application of Petri Net modeling on an Accounting System. It will concentrate on the modeling methodology as applied in Document 1.

5. Derby, S., "Test Case 1 for Structured Assessment Approach (SAA)", ADA, November 27, 1978.

   This memo outlines responsibilities and commitments for both LLL and ADA personnel in the analysis of Facility X. The memo forces a tight time line.

6. Sacks, I. J. and Patenaude, C. J., "Interim Level 3 Analysis", LLL, MC 78-1125-D, November 7, 1978.

   This paper describes the interim status of Level 3 analysis of the SAA, in relation to the Level 3 pre-production computer codes.

7. Freeman, D., "Status of Routines Prob and NRC DB", LLL, MC 78-1109-D, October 20, 1978.

   This paper describes the status of computer programs designed to establish LSD input reliability data (PROB) and NRC data base information (NRCDB).

8. Sacks, I. J., "Techniques for the Determination of Potential Adversary Success With Tampering (Level 4.1)", MC 78-928-D, LLL, October 17, 1978.

   This paper describes the use of Petri Nets to model adversary tampering. Boolean logical expressions which are equivalent to the Petri Net structure are used to describe adversary access to system components. Solutions from the manipulation of the Boolean equations are described. The paper describes basically what happens at the Level 4.1 SAA analysis.

9. Parziale, A. A., "Analysis of a Safeguard Detection and Signal Transmission System as a Stochastic Process, Toward a Potential System Effectiveness Measure", LLL, MC 78-1123, October 10, 1978.

   This paper raises some new questions with regard to safeguard system reliability and availability. Partial solutions are suggested in an attempt to reveal how they work into the SAA. Several topics are discussed and studied. Adversary types are described.

10. Sacks, I. J., "Implementation of Mode for Levels 1, 2, and 3 and Other Questions", MC 78-1049-D, October 9, 1978.

    This is a correspondence memo describing current views of how to model facility operational modes and ramifications relative to Levels 1, 2, and 3 of SAA.

11. Patenaude, S., "Questions About Implementation of Mode Information", LLL, MC 78-992-D, October 6, 1978.

    (This memo initiated Document 10.)

12. Patenaude, C. J., "Monitor Preprocessor", LLL, MC 78-947-D, October 2, 1978.

    This memo describes the status of Level 1 computer codes.

13. Meeting Notes, "Review of Levels 1-4", October 4, 1978.

    These are notes from a meeting held to review all analysis levels of the SAA. They lack text and may be difficult to understand.

14. Siljak, D., "On Structural Properties of MC&A Systems", LLL, MC 78-998, September 29, 1978 and MC 78-758-D, August 1978.

    This paper describes the position of a consultant. The position expressed is not unlike the Level 1 SAA position, but is expressed from a different canonical form point of view.

15. Derby, S., "User Experience with the BKY Computer", LLL, MC 78-1038, September 28, 1978.

    This memo describes the difficulties and delays encountered by the ADA project team in developing and in running the EVENT code on the BKY computer operated by Lawrence Berkeley Laboratory Computer Center.

16. Rice, T. and Derby, S., "Characterization of Analytical Procedures to Calculate the Probability of Successful Diversion of SNM", LLL, MC 78-1037, September 25, 1978.

    This memo briefly reviews the various approaches and approximations to the calculation of the probability of detection by a safeguards system. Related to Level 1 and 3 of the SAA.

17. Derby, S., "Procedure for Combining Utility Networks with Signal Flow Networks", September 22, 1978.

    This memo describes the procedure used in the EVENT code to develop a Boolean expression from unit models, utility paths, and detection signal paths. The EVENT code is used at Levels 2 and 3 of the SAA.

18. Sacks, I. J. and Parziale, A. A., "Unit Models", MC 78-884, LLL, September 20, 1978.

    This memo describes the necessary concept of unit models for describing and modeling safeguard system components. Unit models were found to be necessary to eliminate ambiguity in the analyses. Unit models are used at Level 2.

19. Derby, S., "Implications from Two Definitions of the Level 3 Criterion", September 22, 1978.

    This memo reports on a modeling issue in the definition of a Level 3 criterion. The issue was revealed when two proposed sensitivity studies, calculated by EVENT, gave separate measures of the system effectiveness. The definitions given in Section II of this report (Document 3) were derived from considerations explained in the memo.

20. Sacks, I., "Contents of Subroutine PROB", MC 78-689, LLL, August 15, 1978.

This paper describes the basic contents of the computer subroutine called PROB designed to contain and display and alter LSD system component reliability data. PROB is designed to be interactive with the NRC analyst.

21. Parziale, A.A., "Misinformation Systems", LLL, MC 78-648, August 2, 1978. (Position - Rationale.)

This memo displays a basic concern about eventual assessment outputs to the NRC analyst. It was designed to increase people's awareness of the problem of ill-defined assessment outputs, especially with respect to such irrelevant information.

22. Peterson, J., "Petri Nets", Computing Surveys, Vol. 9; No. 3, received July 20, 1978. Published September 1977.

This paper serves as an excellent tutorial reference on Petri Nets, which are a fundamental modeling tool used in all Level 4 (4.1, 4.2, 4.3) tampering analysis. We include it here for those of you who may not be familiar with Petri Nets.

23. Sacks, I. J., "ADA Work Plans for Level II and Level III Analysis", LLL, MC 78-551, July 12, 1978.

This paper describes ADA's work plan for SAA Level II and Level III analysis development.

24. Rice, T. and Derby, S., "ADA Draft Work Plan", July 7, 1978.

This is a draft work plan from ADA, draft of 23.

25. Parziale, A. A., "Modeling Adversary Tampering of a Safeguards System with a Petri Net", LLL, MC 78-514, June 30, 1978.

This paper, motivated by Document 33, lays down the fundamental modeling philosophy taken in the SAA Level 4 tampering analysis using Petri Nets. The link between Level 4 modeling and LSD input adjacency data is established.

26. Sacks, I. J., "Computation of Network Reliability from Directed Graph", LLL, MC 78-478, June 23, 1978.

This paper addresses the issue of reliability computations from a directed graph. Potential problems and pitfalls are delineated. (Related to Level 2.)

27. Sacks, I. J., "Adversary Success Probabilities with Multiple Attempts", LLL, MC 78-473, June 22, 1978.

    This memo identifies the necessity of recognizing and modeling adversary multiple attempts. The paper had its greatest impact on MC program macro modeling, but the multi-threat concept has been folded into the SAA as well.

28. Derby, S., "Algorithm for Calculating Event Probability from a Directed Graph", June 14, 1978.

    This memo outlines an approach to a calculation for probability of detection. The algorithm in the code EVENT grew out of this initial definition of the problem. (Level 2 and 3 computer code.)

29. Parziale, A. A., "Analysis of Utility Networks and Their Significance in Identifying Vulnerability in a Safeguard Communication System", LLL, MC 78-403, May 31, 1978.

    This paper stresses the importance of all safeguard system components including ancillary utility systems (ELEC.) which support safeguard detection systems - deterministic vulnerability analysis is performed to identify important system components.

30. Rice, T., "Proposal for Analysis - Algorithms for Safeguards Assessment", ADA, May 11, 1978.

    This paper outlines a proposal by ADA to do algorithmic development in the SAA. Initial phase point of view.

31. Parziale, A. A., "Determining Dominant Paths in a Network and Their Significance in Analyzing the Safeguarding of Adversary Area Traversal", LLL, MC 78-301, May 2, 1978.

    This paper describes the concept of "dominance" used in the SAA to reduce and control the amount of analysis done. The dominance concept is applied in more than one way in the SAA.

32. Parziale, A. A., "Analysis of Network Connectivity/Separability, and the Structural Integrity of a Safeguards Communication System", LLL, MC 78-289, April 28, 1978.

    This paper addresses the basic graph theory concepts of connectivity and separability and how they can be applied in a meaningful way to safeguard information flow networks. Vulnerability analysis is entertained. Information transmission networks are explicitly recognized as LSD input toward a complete analysis.

33. Siljak, D. (1977 c), "Petri Nets as Models of Safeguard Systems", Internal Report, LLL, presented April 1978.

This paper introduced the Petri Net model to the MC&A program, and is responsible for initiating further work in this area as described in Document 25.

34. Sacks, I. J., Parziale, A. A., Schrot, M., and Long, J., "A Structured Approach to the Assessment of Material Control and Accounting Systems", LLL, MC 78-203, March 1978.

This key paper describes the SAA Levels 1 through 3, as well as the LSD content and form necessary for the analyses. This paper should be read first.

35. Parziale, A. A., "Personnel Requirements for Continuation of the Alternate Approach", LLL, MC 78-123, February 28, 1978.

This memo describes personnel requirements for further development of the SAA.

36. Sacks, I. J., "Sequence Categories", LLL, MC 78-14, January 13, 1978.

This paper describes basic adversary sequence categories adopted by the aggregated macro model. It had an impact on the design of the SAA, which entertains distinct adversary type (categories).

37. Schrot, M., "Presentation Notes on a Technique for Evaluating Safeguard Systems", December 1977.

These are notes from a presentation designed to describe the SAA at its early stages.

38. Long, J., Schrot, M., and Parziale, A. A., "An Approach to Evaluation of a Material Control System", MC 78-293, November 1977.

This key paper presents many of the basic concepts and approaches, expressed again in Document 34, and which are still a fundamental part of the SAA as it exists to date. Many weeks of work went into development toward this paper which expresses seemingly simple but critical ideas.

39. Weissenberger, S., (1977), "Simple Measures of Vulnerability of Networks", Internal Report LLL, MC 77-486.

This paper suggests simple measures of vulnerability from network analysis under assumptions.

40. Parziale, A. A., "Record of Monthly Reports", September 1977 through January 1979.

This document is a list of handwritten monthly reports of progress from September 1977 to January 1979. The sequence of SAA development events is delineated, but not in great detail.

41. Siljak, D., Patenaude, C. J., Parziale, A. A., Sacks, I. J., "Processing Large Monitored Graphs", LLL, MC 79-242-D, April 23, 1979.

This memo addresses the problem of processing networks which contain very large numbers of nodes. A procedure for condensing these graphs is presented and has been implemented in the preprocessor for the SAA core codes.

42. Sacks, I. J., Patenaude, C. J., "Level 1.5 - Weak Collusion Test For the Structured Assessment Approach", LLL, MC 79-338, July 6, 1979.

This memo describes the sample collusion test incorporated into the SAA following Level 1 processing. Level 1.5 performs an analysis similar to the analysis performed by the Science Applications, Inc., developed Matrix Analysis of the Insider Threat (MAIT).

43. Patenaude, C. J., Sacks, I. J., "Revised Level II/III Flowcharts", LLL, MC 79-296-D, May 30, 1979.

This memo describes the flowcharts used to develop the computer codes for Levels 2 and 3 of the SAA.

44. Parziale, A. A., "Safeguard System False Alarm Rate", LLL, MC 79-358-D, July 13, 1979.

This memo describes the calculation of MC system false alarm rate used in the SAA.

45. Patenaude, C. J., "Bounding on the Probability Code", LLL, MC 79-364-D, July 20, 1979.

This memo describes the approximation technique used in the SAA to keep the computations required by the Level 2 analysis within a reasonable bound.

46. Sacks, I. J., "Level 3.5 Second Phase Collusion Test", LLL, MC 79-313-D, June 14, 1979.

This memo describes the collusion analysis which will be used on the complete MC&A network for any Monitor Target Set.

47. Renard, P. A., "CLAMOR - A Preliminary Report", LLL, MC 79-96, January 31, 1979.

This memo describes the computer code which processes the Petri Nets of Level 4.

48. Patenaude, C. J., Parziale, A. A., Sacks, I. J., Ross, D. J., "SAA User Interaction for October, 1979", LLL, MC 79-314-D, June 15, 1979.

This memo was intended as a working document toward the end of defining the interface between the NRC License Reviewer and the SAA codes.

49. Renard, P. A., "Warshall on Symmetric Matrices", LLL, MC 79-207, February 15, 1979.

This memo describes the Warshall algorithm which is used to compute transitive closure and its specification to adjacency matrices which are symmetric.

50. Maimoni, A., "Uses of Graphs For Describing Procedures", LLL, S 79-246-D, April 23, 1979.

This report describes the technique used in the SAA to model procedures such as the use of a contraband detector at the entrance to a nuclear facility.

51. Sacks, I. J., "Partial Condensation Rules I", LLL, MC 79-126, March 12, 1979.

This memo describes a set of rules which will partially condense some networks of both the uncontrolled and controlled type.

52. Renard, P. A., "Clarification of Partial Condensation Rules", LLL, MC 79-127, February 26, 1979.

This memo clarifies the rules described in item 51 above.

53. Sacks, I. J., Parziale, A. A., Renard, P. A., "The Modeling of Procedures", LLL, MC 78-1374-D, February 1, 1979.

This memo describes the modeling technique used for Level 4 analysis.

54. Patenaude, C. J., Sacks, I. J., "Dominance Tests", LLL, MC 79-89, February 6, 1979.

This memo describes the implementation of the dominance tests into the SAA Level 1 code.

LLNL* 1981/12

☆ U.S. GOVERNMENT PRINTING OFFICE: 1982 – 372-873

NRC FORM 335
(7-77)

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

1. REPORT NUMBER *(Assigned by DDC)*
NUREG/CR-1233, Vol. IV
UCID-18146

4. TITLE AND SUBTITLE *(Add Volume No., if appropriate)*
The Structured Assessment Approach
Version 1
Computational Analysis Package

2. *(Leave blank)*

3. RECIPIENT'S ACCESSION NO.

7. AUTHOR(S)
A. A. Parziale, C. J. Patenaude, P. A. Renard,
D. W. Freeman, D. J. Ross (LLNL) I. J. Sacks (AIP)

5. DATE REPORT COMPLETED

MONTH	YEAR
October	1979

9. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS *(Include Zip Code)*
Lawrence Livermore National Laboratory
P. O. Box 808
Livermore, CA 94550

DATE REPORT ISSUED

MONTH	YEAR
April	1982

6. *(Leave blank)*

8. *(Leave blank)*

12. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS *(Include Zip Code)*

Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555

10. PROJECT/TASK/WORK UNIT NO.

11. CONTRACT NO.
NRC FIN A0115

13. TYPE OF REPORT
NUREG

PERIOD COVERED *(Inclusive dates)*

15. SUPPLEMENTARY NOTES

14. *(Leave blank)*

16. ABSTRACT *(200 words or less)*

A methodology, called the Structured Assessment Approach (SAA), has been developed to assess the effectiveness of material control and accounting (MC&A) safeguards systems at nuclear fuel cycle facilities. The methodology has been refined into a computational tool, the Version 1 analysis package, that has been used first to assess a hypothetical nuclear fuel cycle facility and more recently to assess operational nuclear plants.

The Version 1 analysis package is designed to analyze safeguards systems that prevent the diversion of Special Nuclear Material from nuclear fuel cycle facilities and to provide assurance that diversion has not occurred.

17. KEY WORDS AND DOCUMENT ANALYSIS

17a. DESCRIPTORS

17b. IDENTIFIERS/OPEN-ENDED TERMS

18. AVAILABILITY STATEMENT
Unlimited

19. SECURITY CLASS *(This report)*
Unclassified

20. SECURITY CLASS *(This page)*
Unclassified

21. NO. OF PAGES

22. PRICE
$

NRC FORM 335 (7-77)

131

BEST DOCUMENT AVAILABLE